

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ**

«На правах рукопису»
УДК 004.08

«До захисту допущено»
Завідувач кафедри СПСКС

_____ В.П.Тарасенко
(підпис) (ініціали, прізвище)
“ ” _____ 2018р.

**Магістерська дисертація
на здобуття ступеня магістра**

зі спеціальності 123 Комп'ютерна інженерія
Системне програмування

на тему МОДИФІКАЦІЯ І ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖІ ХЕББА
Виконав: студент II курсу, групи зКВ-71мп
(шифр групи)

Чухліб Юрій Валентинович
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник к.т.н., доц.каф.СПСКС Боярінова Ю.Є.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія

Системне програмування

ЗАТВЕРДЖУЮ

Зав. кафедри СПСКС

_____ В.П.Тарасенко
(підпис) (ініціали, прізвище)

«__» _____ 2018р.

**ЗАВДАННЯ
на магістерську дисертацію студента
Чухліб Юрія Валентиновича**

1. Тема дисертації: МОДИФІКАЦІЯ І ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖІ ХЕББА,

науковий керівник дисертації: к.т.н., доцент каф.СПСКС Боярінова Ю.Є,
затверджені наказом по університету від «09» листопада 2018 р. № 4138-с.

2. Термін подання студентом дисертації: 10 грудня 2018 р.

3. Об'єкт дослідження: методи та алгоритми навчання нейронних мереж.

4. Предмет дослідження: метод та алгоритм Хебба обробки сигналів і даних у нейронних мережах.

5. Перелік завдань, які потрібно розробити:

- розглянути основні методи навчання нейронних мереж;
- провести порівняльний аналіз основних методів навчання;
- дослідити метод навчання Хебба;
- розробити та описати етапи обраного алгоритму на основі навчання Хебба та описати його модифікацію;
- програмно реалізувати розроблений модифікований алгоритм навчання нейронної мережі;
- провести експерименти і оцінити роботу оптимізованого алгоритму.

6. Перелік ілюстративного матеріалу:

– Презентація.

7. Перелік публікацій: «Захист інформації в корпоративних інформаційних системах», XI наукова конференція молодих вчених «Прикладна математика та комп'ютинг» ПМК-2018-2 (Київ, 14-16 листопада 2018 р.); V Міжнародна науково-технічна конференція «Сучасні тенденції впровадження інформаційних систем керування галузями діяльності» (Київ, 22-23 листопада 2018 р.).

8. Дата видачі завдання 5 вересня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ознайомлення з предметною галуззю	17.12.2017	
2	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	05.02.2018	
3	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	15.03.2018	
4	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	27.06.2018	
5	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	12.09.2018	
6	Проведення наукового дослідження; робота над третім розділом магістерської дисертації; підготовка матеріалів доповіді на конференції «Прикладна математика та комп'ютинг» ПМК-2018-2	03.10.2018	
7	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу	16.10.2018	
8	Оформлення текстової і графічної частини магістерської дисертації	10.11.2018	
9	Попередній розгляд магістерської дисертації на кафедрі	26.11.2018	

Студент _____

Чухліб Ю. В.

Науковий керівник дисертації _____

Боярінова Ю.Є.

РЕФЕРАТ

Актуальність теми.

З кожним роком людство ставить перед собою все більш складні задачі. Для вирішення цих задач створюються все більш складні та потужні алгоритми, які зможуть це зробити набагато швидше та ефективніше, ніж людина. Але, часом, перед нами постають нелінійні задачі, коли набір вхідних даних кожен раз різний, на вхід приходять дані, які потрібно спочатку фільтрувати, а тільки потім вирішувати поставлену задачу. Для такого типу задач необхідно розробити алгоритм, який вміє сумніватися, який розуміє, що не може бути однозначної відповіді і який спроможний знаходити цю відповідь. Наприклад, це можуть бути задачі відстеження фроду при банківських операціях, аномальні показники лічильників у літаку або на атомних станціях. Це можуть бути задачі по розпізнаванню образів з фото або відео даних. Для таких класів задач створюють алгоритми на основі так званого штучного інтелекту. Ці алгоритми спроможні працювати кожен раз з різним набором даним, і так само ефективно при цьому вирішувати поставлену задачу. З часом, якщо набір вхідних даних буде змінюватись, такий алгоритм спроможний адаптуватись до цих змін, не перестаючи так само ефективно працювати. Але для цього потрібно навчити алгоритм та показати йому, яку саме відповідь він має знайти, опираючись на вхідні дані. Проблема в тому, що набори вхідних даних змінюються занадто доволі швидко, а алгоритми навчання доволі повільні, тому необхідно підібрати достатньо швидкі алгоритми, щоб наш штучний інтелект міг так само швидко навчатись та пізніше ефективно адаптуватись до змін вхідних даних.

Об'єктом дослідження є методи та алгоритми навчання нейронних мереж.

Предметом дослідження Предметом дослідження є метод та алгоритм Хебба обробки сигналів і даних у нейронних мережах.

Мета роботи: Метою роботи є створення високоефективних інформаційних технологій обробки та перетворення зображень і даних на базі архітектури системних обчислювальних середовищ та нейронних мереж

для підвищення точності функціонування сучасних нейромереж як в режимі відтворення, так і в режимі навчання.

Наукова новизна:

1. Запропоновано модифікований спосіб навчання, який базується на алгоритмі Хебба.
2. Виконано порівняльний аналіз розробленого способу з існуючим стандартним способом для вирішення задачі розпізнавання рукописних цифр.

Практична цінність отриманих в роботі результатів полягає в тому, що розроблений модифікований спосіб навчання є більш точним, ніж стандартний.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, трьох розділів та висновків.

У вступі подано узагальнену оцінку сучасного стану розвитку алгоритмів, обґрунтовано актуальність виконаного дослідження, дано загальну характеристику роботи, поставлено мету та задачу дослідження, і наведено практичну цінність роботи.

У першому розділі розглянуто основні відмінності алгоритмів на основі штучного інтелекту, розглянуто існуючі методи навчання цих алгоритмів, їх принципи роботи та особливості, ключові недоліки та переваги.

У другому розділі розглянуто принцип роботи навчання та алгоритму Хебба.

У третьому розділі описано деталі реалізації базового та модифікованого алгоритму, створено програмний продукт на базі цих алгоритмів, зроблену порівняльний аналіз.

У висновках представлені результати проведеної роботи.

Робота представлена на 80 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: штучний інтелект, нейронна мережа, метод Хебба.

ABSTRACT

Actuality of topic.

Every year, mankind sets an increasingly complex task. To solve these problems, increasingly sophisticated and powerful algorithms are created that can do this much faster and more efficiently than humans. But sometimes, we face nonlinear problems, when the set of input data is different each time, the input comes with the data that you must firstly filter, and only then solve the task. For this type of task, you need to develop an algorithm that you can doubt, which understands that there can not be an unambiguous answer and who is able to find this answer. For example, it may be the task of tracking the frode in banking operations, abnormal counters in the aircraft or at nuclear power plants. This may be tasks for recognizing images from photo or video data. For such pazzles, tasks are created by an algorithm based on the so-called artificial intelligence. These algorithms are able to work each time with a different set of data, and also effectively solve this problem. Over time, if the input set is to be changed, such an algorithm will be able to adapt to these changes, without interrupting the same efficient operation. But to do this, you need to teach the algorithm and show him which answer it should find based on the input. The problem is that the input data changes too quickly, and the learning algorithms are quite slow, so you need to pick up a fast enough algorithm to allow our artificial intelligence to learn as quickly as possible and then adapt effectively to the changes in the input data.

The object of the study is the methods and algorithms for teaching neural networks.

Subject of research is the Hebbian method and algorithm for processing signals and data in neural networks.

Purpose: The purpose of the work is to create high-performance information technology for the processing and transformation of images and data based on the architecture of system computing environments and neural networks to improve the accuracy of the functioning of modern neural networks in both playback mode and in training mode.

Scientific novelty:

1. A modified method of learning is proposed, which is based on Hebbian method.
2. A comparative analysis of the developed method with the existing standard method for solving the problem of recognition of handwritten digits is performed.

The practical value of the results obtained in the work is that the developed modified learning method is more precise than the standard one.

Structure and scope of work. The master's dissertation consists of an introduction, three sections and conclusions.

In the introduction a generalized estimation of the current state of algorithms development is presented, the relevance of the performed research is substantiated, a general description of the work is given, the purpose and the task of the research are stated, and the practical value of the work is given.

The first chapter deals with the main differences in algorithms based on artificial intelligence, examines the existing methods of teaching these algorithms, their principles of operation and features, key disadvantages and advantages.

The second chapter deals with the teaching principle and Hebb's algorithm.

The third section describes the details of the implementation of the basic and modified algorithm, created a software product on the basis of these algorithms, made a comparative analysis.

The conclusions are the results of the work.

The work is presented on 80 sheets, contains a link to the list of used literary sources.

Key words: artificial intelligence, neural network, Hebbian method.

ЗМІСТ

ВСТУП	10
1. АНАЛІЗ НЕЙРОМЕРЕЖ	12
1.1. Поняття нейромереж та їх використання	12
1.2. Поняття нейромережі Хебба	23
2. ДОСЛІДЖЕННЯ ПОБУДОВИ НЕЙРОМЕРЕЖ	32
2.1. Математична модель нейромережі	32
2.2. Дослідження складових та етапів побудови нейромережі	38
2.3. Навчання нейромережі методом Хебба	Ошибка! Закладка не определена.
3. РОЗРОБКА НЕЙРОМЕРЕЖІ ХЕББА	58
3.1. Постановка задачі та опис алгоритму	60
3.2. Застосування розробленої нейромережі Хебба	80
ВИСНОВКИ	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87
ДОДАТКИ	

Перелік умовних скорочень, позначень, термінів

ВРО – внутрішні рецидивні одиниці

КУ – компонентне уявлення

МБ – машина Больцмана

МДП - механізмами довгострокового потенціювання

МХ – мережа Хопфілда

НМ – нейронна мережа

ПХ – перетворення Хаффа

УРНМ - узагальнена регресійна нейронна мережа

ШНМ – штучна нейронна мережа

CNN – конволюційна нейронна мережа

IDE – інтегроване середовище розробки

RNN – рекурентна нейронна мережа

ВСТУП

Штучні нейронні мережі є ключовим елементом деяких найбільш успішних алгоритмів машинного навчання. Розвиток нейронних мереж був ключовим для навчання комп'ютерів, щоб мислити та розуміти світ таким чином, як люди це роблять.

Нейронна мережа має вхідні і вихідні нейрони, які пов'язані ваговими синапсами. Ваги впливають на те, наскільки велика частина прямого розповсюдження проходить через нейронну мережу. Ваги можуть бути змінені під час процесу зворотного розповсюдження - це процес, де нейронна мережа навчається. Цей процес прямого розповсюдження та зворотного поширення ведеться ітеративно на кожному об'єкті даних в наборі навчальних даних. Що більший розмір набору даних і чим більше встановлена різноманітність даних, тим більше ймовірність, що нейронна мережа буде краще навчатись і отримає краще прогнозування виходів. Для адаптації, налаштування або навчання ваг зв'язків кожного нейрона може бути використане декілька методів.

Запропоновано розглянути один з них, який отримав назву "правило Хебба". Під час дослідження механізму функціонування центральної нервової системи, Хебб припустив, що навчання кожного нейрона та їх поєднань відбувається шляхом підсилення зв'язків між нейронами, у яких однакова активність на одному проміжку часу.

Наприклад, в системах біологічного походження таке припущення справедливе досить рідко і всі види навчання не вичерпуються, однак при навчанні нейронних мереж з одним шаром з біполярними сигналами це досить ефективно.

Іншим ефективним напрямом є побудова інваріантів на базі перетворень Хемінга. Раніше ці перетворення з успіхом застосовувалися в оптичних системах. Незабаром були розроблені швидкі ефективні алгоритми

їх застосування для цифрових зображень. Шляхом застосування спектральних перетворень можна формувати інваріантні ознаки, а також оцінювати величини перетворень. Безсумнівним достоїнством спектральних методів є висока перешкодозахищеність до адитивних перешкод, а також можливість вибіркового використання найменш перекручених компонентів спектра для розпізнавання на базі процедур багаторазового прийняття рішень. До недоліків можна віднести збільшення часу і витрат пам'яті, викликані ускладненням обробки через застосування перетворень.

Вагомий внесок у розвиток теорії створення надвисокопродуктивних засобів обробки інформації в системних та нейронних середовищах внесли такі вчені Айвазян С.А., Бухштабер В.М., Єнюков И.С., Мешалкін Л.Д., Алієв Р.А., Абдикєв Н.М., Шахназаров та ін. [1,2]

1. АНАЛІЗ НЕЙРОМЕРЕЖ

1.1. Поняття нейромереж та їх використання

Чому ми потребуємо алгоритми машинного навчання? Машинне навчання потрібне для завдань, які занадто складні для кодування алгоритмів людьми безпосередньо. Деякі завдання настільки складні, що люди часами просто не можуть побачити усі нюанси поставленої задачі, до якої потрібно написати алгоритм. Тому замість цього ми пропонуємо великий обсяг даних для алгоритму машинного навчання, і алгоритм може працювати, досліджуючи ці дані та шукаючи модель, яка досягне того, що програмісти не можуть досягти.

Дуже важко писати програми, які вирішують такі проблеми, як розпізнавання 3-мірного об'єкта з нової точки зору в нових умовах освітлення в затіненій сцені. Ми не знаємо, яку програму писати, бо ми не знаємо, як це робиться в нашому мозку. Навіть якщо б ми мали гарне уявлення про те, як це зробити, програма може бути жахливо складною.

Важко написати програму для обчислення ймовірності того, що операція з кредитною картою є шахрайською. Там не може бути будь-яких правил, які є простими та надійними. Нам потрібно об'єднати дуже велику кількість слабких правил. Шахрайство - це рухома мета, але програма потребує постійного зміни.

Тому використаємо метод машинного навчання: замість написання програми вручну для кожного конкретного завдання, ми збираємо багато прикладів, які вказують правильний вихід для даного вводу. Алгоритм машинного навчання потім приймає ці приклади і видає програму, яка виконує роботу. Програма, розроблена алгоритмом машинного навчання, може виглядати значно відмінною від звичайної написаної вручну програми. Вона може містити мільйони чисел та параметрів. Якщо ми зробимо це

правильно, програма працює для нових випадків, а також тих, на яких ми навчалися. Якщо дані змінюються, програма може також змінюватися під час тренування нових даних. Слід зазначити, що величезні обсяги обчислень зараз дешевші, ніж платити комусь для написання спеціальної програми.

З огляду на це, деякі приклади завдань, які найкраще вирішуються в процесі машинного навчання, включають:

- Розпізнавання шаблонів: Об'єкти в реальних сценах, особистість особи або вираз обличчя, розмовні слова.
- Визнання аномалій: незвичні послідовності транзакцій з кредитними картками, незвичні моделі відліків датчиків на атомній електростанції.
- Прогнозування: майбутні ціни на акції або курси обміну валют, які фільми вам сподобаються.

Що таке нейронні мережі? Нейронні мережі - це клас моделей в загальній літературі для машинного навчання. Нейронні мережі - це специфічний набір алгоритмів, який змінив область машинного навчання. Вони надихаються біологічними нейронними мережами, і сучасні діючі так звані глибокі нейронні мережі довели свою роботу дуже добре. Самі нейронні мережі є наближенням загальної функції, тому їх можна застосувати до буквально практично будь-якої проблеми машинного навчання, коли проблема полягає у вивченні складного відображення від входу до вихідного простору.

Ось три причини, навіщо потрібно розуміти суть нейронних обчислень:

- Щоб зрозуміти, як мозок насправді працює: це дуже велика і дуже складна система і зроблена з речовини, яка помирає, коли ви дотикаєтесь. Отже, нам потрібно використовувати комп'ютерне моделювання.
- Щоб зрозуміти стиль паралельного обчислення, натхненний нейронами та їх адаптивними зв'язками: це кардинально інший стиль від послідовних обчислень.

- Для вирішення практичних завдань, використовуючи нові алгоритми навчання, натхненні мозком: навчальні алгоритми можуть бути дуже корисними, навіть якщо вони не є тим, як мозок насправді працює.

Пропоную розглянути 10 основних архітектур нейронних мереж та основні принципи їх роботи.

1. Персептрон. Це проста модель біологічного нейрона в штучній нейронній мережі. Персептроном також називається ранній алгоритм для нагляду за вивченням бінарних класифікаторів.

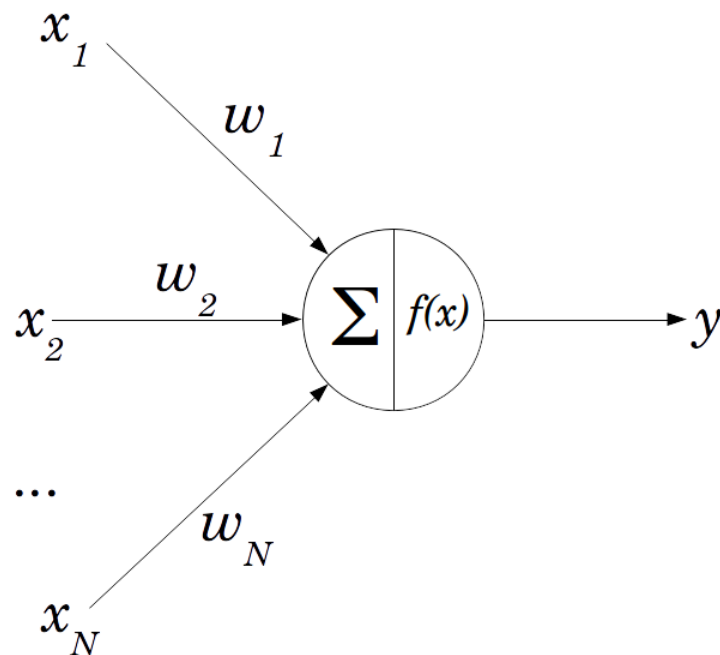


Рисунок 1.1 – Загальна структура персептрона

Алгоритм персептрон був призначений для класифікації візуальних входів, категоризації предметів на один з двох типів і розділення груп з лінією. Класифікація є важливою частиною машинного навчання та обробки зображень. Алгоритми машинного навчання дозволяють знаходити та класифікувати моделі за допомогою багатьох різних засобів. Алгоритм персептрон класифікує шаблони та групи, знаходячи лінійне відокремлення між різними об'єктами та шаблонами, отриманими через числовий чи візуальний вхід.

Алгоритм персептрон був розроблений в Корнельській авіаційній лабораторії в 1957 році, який фінансується Управлінням військово-морських досліджень Сполучених Штатів. Це був перший крок, який планувався для впровадження машини розпізнавання зображень. Машина, яка мала назву Март-1 Персептрон, фізично складалася з 400 фотоелементів, з'єднаних з персептронами, ваги яких були записані в потенціометрах, що регулювалися електродвигунами. Машина була однією з перших штучних нейронних мереж, будь-коли створених.

У той час передбачалося, що персептрон буде дуже значним для розвитку штучного інтелекту (ШІ). Хоча на персептрон були покладені великі надії, дуже скоро було виявлено також деякі технічні обмеження. Одношарові персептрони можуть виділяти класи тільки в тому випадку, якщо вони є лінійними. Пізніше було знайдено, що за допомогою декількох шарів персептрони можуть класифікувати групи, які не є лінійно відокремленими, що дозволяє їм вирішувати проблеми, алгоритми одного шару не можуть вирішити.

2. Конволюційні (згорткові) нейронні мережі. У 1998 році Янн ЛеКун та його співробітники розробили ефективний засіб для розпізнавання рукописних цифр під назвою LeNet. Вона використовувала зворотне поширення в прямої мережі з багатьма прихованими шарами, безліччю карт повторюваних вузлів у кожному шарі, об'єднанням результатів найближчих повторюваних вузлів, широкою мережею, яка може справлятися одразу з кількома символами, навіть якщо вони перекриваються, і розумним способом навчання повної системи. Пізніше це було формалізовано під назвою згорткових нейронних мереж (CNNs). Цікавий факт: ця мережа була використана для читання ~ 10% чеків у Північній Америці.

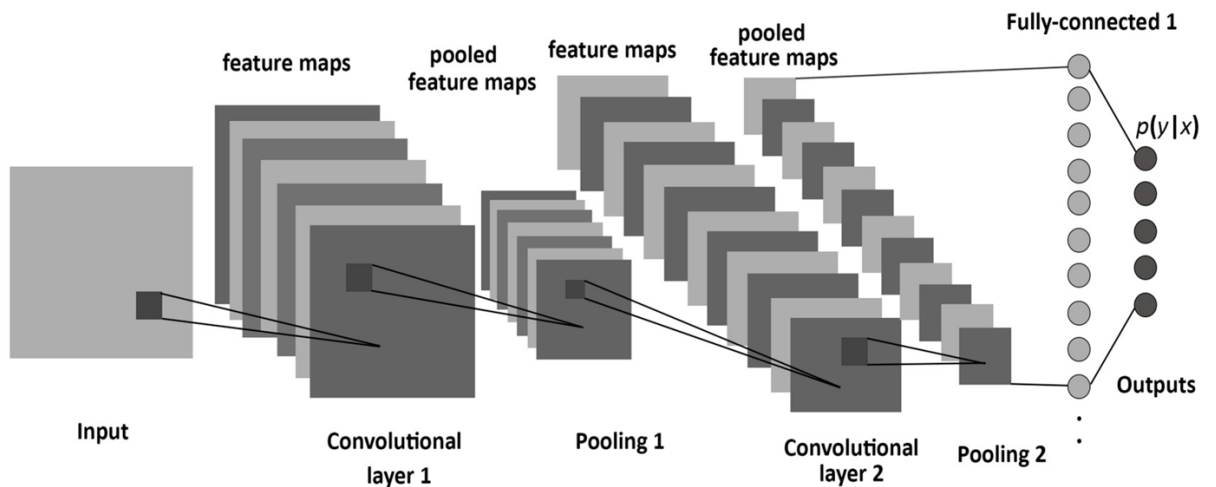


Рисунок 1.2 – Загальна схема згорткової нейронної мережі

Конволюційні нейронні мережі сильно відрізняються від більшості інших мереж. Вони використовуються для обробки зображень, але можуть також використовуватися для інших типів входів, таких як аудіо. Типовим випадком використання CNN є передача мережових зображення, де в наслідок мережа класифікує їх на типи. CNN, як правило, має вхідний «сканер», який зазвичай не призначений для аналізу всіх навчальних даних одночасно. Наприклад, щоб ввести зображення розміром 100 x 100 пікселів, ви не хочете, щоб шар містив 10 000 вузлів. Швидше за все, ви створюєте шаблон сканування буде мати 10 x 10 вузлів, на які буде перенаправлене 10 x 10 пікселів зображення. Після того, як цей вхід буде пройдено, можна буде подати наступні 10 x 10 пікселів та перенести сканер в один піксель вправо.

Ці вхідні дані потім подаються через згорткові шари замість звичайних шарів, де не всі вузли з'єднуються з усіма вузлами. Кожен вузол стосується лише близьких сусідніх клітин. Такі згорткові шари, як правило, зменшуються, коли стають більш глибокими, в загальному випадку легко подільними факторами входу. Окрім цих згорткових шарів, у них також присутні об'єднані шари. Поєднання є способом фільтрації деталей: загальноприйнята методика накопичення – це максимальне

об'єднання, де ми приймаємо 2 x 2 пікселя і передаємо пікселю найбільшу кількість червоного кольору.

3. Рекурентні нейронні мережі (РНН). Щоб зрозуміти РНН, треба мати короткий огляд моделювання послідовності. При застосуванні машинного навчання до послідовностей найчастіше задачею є перетворення вхідної послідовності у вихідну, яка живе в іншому домені; наприклад, перетворити послідовність звукових тисків у послідовність ідентифікаторів слів. Якщо немає окремої цільової послідовності, ми можемо отримати сигнал навчання, намагаючись передбачити наступний термін у вхідній послідовності. Вихідна послідовність цілі - це вхідна послідовність із переходом на 1 крок. Це здається набагато більш природним, ніж спроба передбачити один піксель у зображенні з інших пікселів. Прогнозування наступного кроку в послідовності має невелику відмінність між контрольованим та безконтрольним навчанням. Він використовує методи, призначені для керованого навчання, але не вимагає окремого викладацького сигналу.

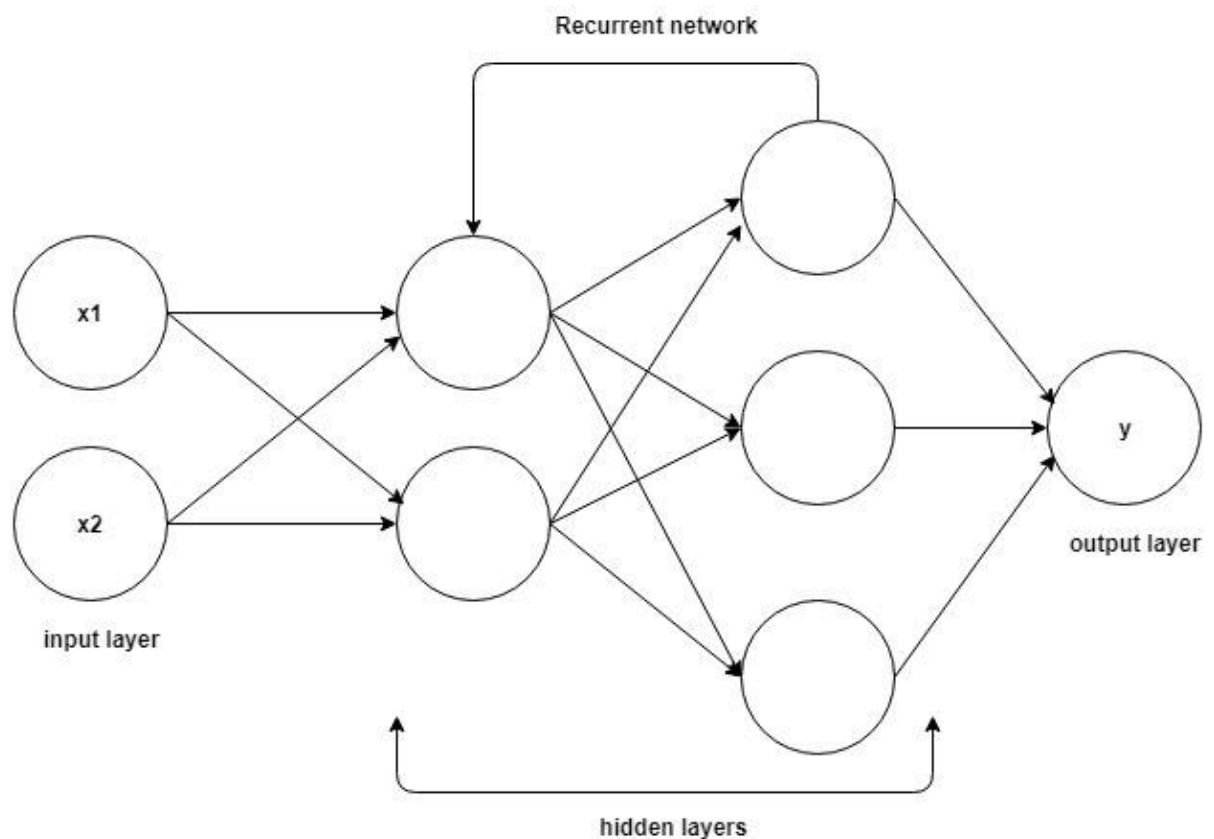


Рисунок 1.3 – Загальна схема рекурентної нейронної мережі

Моделі без пам'яті є стандартним підходом до цього завдання. Зокрема, авторегресивні моделі можуть передбачати наступний крок у послідовності з фіксованого числа попередніх термінів за допомогою "затримки"; і непрямі нервові сітки - генералізовані авторегресивні моделі, що використовують один або більше шарів нелінійних прихованих одиниць. Однак, якщо ми дамо нашій моделі деякий прихований стан, і якщо ми дамо цьому прихованому стану власну внутрішню динаміку, ми отримаємо набагато більш цікаву модель: вона зможе зберігати інформацію в прихованому стані довше. Якщо динаміка шумна, і спосіб, яким вона генерує виходи зі свого прихованого стану, шумні, ми ніколи не зможемо дізнатись про її прихований стан. Найкращий варіант, це зробити висновок про розподіл імовірності через простір векторів прихованого стану. Це висновок можна розглядати лише для двох типів прихованої моделі стану.

RNN дуже потужні, оскільки вони об'єднують 2 властивості: 1) розподілений прихований стан, який дозволяє їм зберігати багато інформації про минуле ефективно; і нелінійна динаміка, яка дозволяє їм оновлювати їх прихований стан складними шляхами. За наявності достатніх нейронів і часу, RNN можуть обчислювати все, що може бути обчислено комп'ютером. Які види поведінки можуть виявляти RNN? Вони можуть коливатися, вони можуть оселитися, щоб визначити пріоритет, вони можуть вести себе хаотично. І вони потенційно можуть навчитися реалізовувати безліч невеликих програм, кожен з яких захоплює область знань і працює паралельно, взаємодіючи між собою та створюючи дуже складні ефекти.

Одна з великих проблем з RNN - це зникнення (або вибух) градієнта, в якій, залежно від використовуваних функцій активації, інформація швидко втрачається з часом. Інтуїтивно це не було б серйозною проблемою, оскільки ці просто ваги, а не нейронні стани, але вага за часом - це фактично місце збереження інформації з минулого; якщо вага

досягає значення 0 або 1 000 000, попереднє стан не буде дуже інформативним. RNN можуть використовуватися у багатьох областях, оскільки більшість форм даних насправді не мають часової шкали (на відміну від звуку або відео), можуть бути представлені як послідовність. Картину або рядок тексту можуть одночасно подавати один піксель або символ, тому залежні від часу ваги використовуються для того, що було раніше в послідовності, а не фактично від того, що трапилось за секунди раніше. Загалом, періодичні мережі - це хороший вибір для просування або заповнення інформації, наприклад, автозавершення.

4. Довга та короткочасна пам'ять. Деякі дослідження вивели рішення для проблеми отримання RNN з пам'яттю, будуючи так звані довгострокові/короткострокові мережі пам'яті (LSTM). Мережі LSTM намагаються боротися з проблемою зникнення / вибуху градієнта шляхом введення воріт і явної комірки пам'яті. У LSTM також є "вхідні ворота", які додають нові елементи до комірки і "вихідні ворота", які вирішують, коли проходити уздовж векторів з клітинки в наступний прихований стан.

З усіма RNN значення, що надходять з X_{train} та $H_{previous}$, використовуються для визначення того, що відбувається в поточному прихованому стані. І результати поточного прихованого стану ($H_{current}$) використовуються для визначення того, що відбувається в наступному прихованому стані.

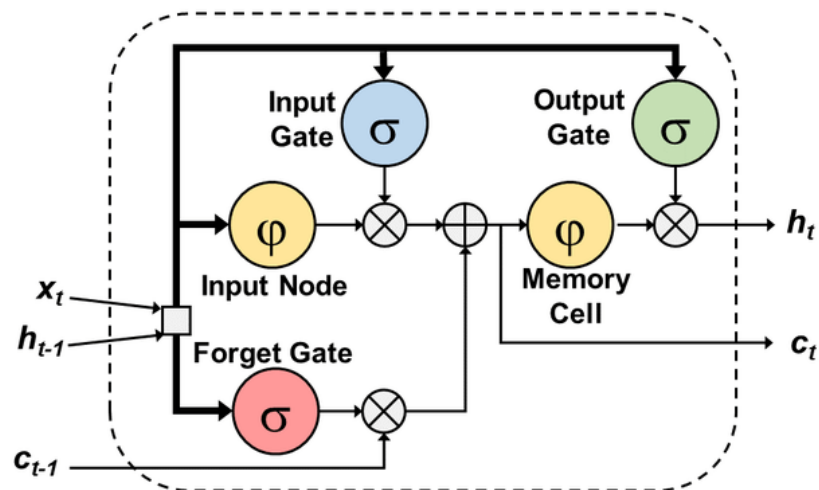


Рисунок 1.4 – Загальна схема блоку LSTM

LSTM просто додають шар клітин, щоб переконатися, що передача прихованої інформації про стан від однієї ітерації до наступної є досить високою. Іншими словами, ми хочемо запам'ятати речі з попередніх ітерацій до тих пір, поки це потрібно, і клітини в LSTM дозволяють це зробити.

5. Відрегульований блок. Внутрішні рецидивні одиниці (GRU) є невеликими варіаціями на LSTMs. Вони беруть X_{train} і $H_{previous}$ в якості вхідних даних. Вони виконують деякі розрахунки, а потім проходять уздовж $H_{current}$. У наступній ітерації $X_{train.next}$ та $H_{current}$ використовуються для додаткових обчислень, тощо. Те, що робить їх відмінними від LSTM, полягає в тому, що GRU не потребує клітинного шару, щоб передавати значення разом. Підрахунки в кожній ітерації гарантують, що значення $H_{current}$, які передаються разом, зберігають велику кількість старої інформації або починають стрибати з великою кількістю нової інформації.

У більшості випадків GRU функціонує дуже подібно до LSTM, причому найбільша різниця між тим, що GRU трохи швидше і простіше запускати (але також трохи менш виразна). На практиці вони, як правило, відміняють один одного, тому що потрібна більша мережа, щоб

відновити деяку виразність, яка в свою чергу скасовує переваги продуктивності. У деяких випадках, коли додаткова експресивність не потрібна, GRU може перевершити LSTM.

6. Мережа Хопфілд. Регулярні мережі нелінійних одиниць, як правило, дуже важко проаналізувати. Вони можуть вести себе по-різному: осідають до стабільного стану, коливаються або дотримуються хаотичних траєкторій, які неможливо передбачити далеко в майбутньому. Щоб вирішити цю проблему, Джон Хопфілд представив Hopfield Net у своїй роботі в 1982 р. «Нейромережі та фізичні системи з виникненням колективних обчислювальних здібностей». Мережа Хопфілд (HN) - це мережа, де кожен нейрон підключається до кожного іншого нейрону; це повністю заплутана тарілка спагетті. Кожен вузол вводиться перед тренуванням, потім приховано під час тренувань та виводу після нього. Мережі навчаються, встановлюючи значення нейронів до бажаного малюнка, після чого ваги можуть бути обчислені. Ваги після цього не змінюються. Після навчання для одного або декількох моделей, мережа завжди збігатиметься з одним із вивчених моделей, оскільки мережа стабільна лише в цих станах.
- Існує ще одна обчислювальна роль для мережі Хопфілда. Замість того, щоб використовувати мережу для зберігання спогадів, ми використовуємо її для побудови інтерпретацій сенсорного вводу. Вхідні дані представлені видимими одиницями, інтерпретація представлена станами прихованих одиниць, а погана інтерпретація представлена енергією.

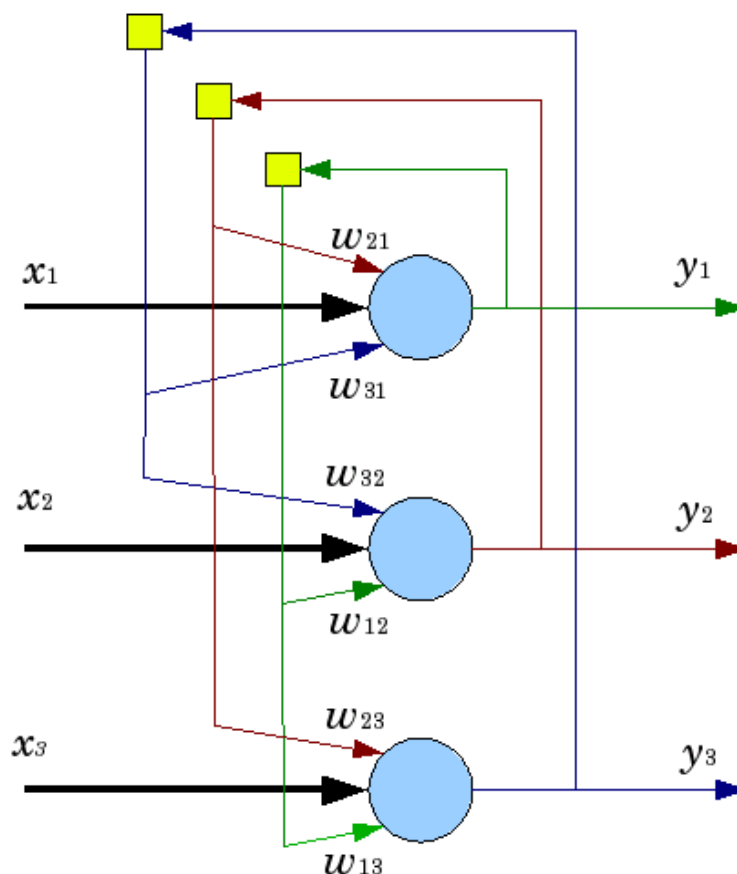


Рисунок 1.5 – Загальна схема нейронної мережі Хопфілда

На жаль, люди показали, що мережа Хопфілда дуже обмежена у своїй якості. Сітка Хопфілда з N одиниць може запам'ятовувати лише $0,15N$ шаблони через так звані хибні мінімуми в його енергетичній функції. Ідея полягає в тому, що оскільки функція енергії є безперервною в просторі її ваг, якщо два локальних мінімуму занадто близько, вони можуть "падати" один в одного, щоб створити єдині локальні мінімуми, які не відповідають будь-яким навчальним зразкам, тоді як забуваючи про два зразки він повинен запам'ятати. Це явище суттєво обмежує кількість зразків, які може вивчати мережа Хопфілда.

7. Машина Больцмана. Машина Больцмана - це тип стохастичної рекурентної нейронної мережі. Її можна розглядати як стохастичний, генеративний аналог мережі Хопфілда. Це була одна з перших нейронних мереж, здатних вивчати внутрішні уявлення, і здатна представляти і вирішувати складні комбінаторні проблеми. Спочатку

представлені Джефрієм Хінтоном і Терренсом Сейновскі в "Навчанні та перепідготовці в машинах Boltzmann" (1986), машини Больцмана дуже схожі на Hopfield Networks, але деякі нейрони відмічені як вхідні нейрони, а інші залишаються "прихованими". Введення нейронів стає вихідними нейронами в кінці повного оновлення мережі. Він починається з випадкових ваг і навчається через зворотне поширення. Порівняно з мережею Hopfield Net, нейрони більшою мірою мають двійкові шаблони активації.

Метою навчання для алгоритму машинного навчання Больцмана є максимізація продукту ймовірностей, які машина Больцмана присвоює двійковим векторам в наборі тренувань. Це еквівалентно максимізації суми значень журналів, які машина Больцмана призначає тренувальним векторам. Це також еквівалентно збільшенню ймовірності того, що ми отримали б точно N випадки навчання, якщо ми зробили наступне: 1) Нехай мережа осідає на своєму стаціонарному розподілі N різного часу без зовнішнього введення; 2) Зразок видимого вектора раз на кожен раз. Ефективна процедура міні-пакетного навчання була запропонована для Boltzmann Machines від Салахутдінова та Хінтона в 2012 році. Для позитивної фази спочатку ініціалізуйте приховані ймовірності на рівні 0,5, потім закріпіть вектори даних на видимі одиниці, а потім оновіть всі приховані одиниці паралельно, аж до конвергенції за допомогою середніх оновлень поля. Після того, як мережа зросте, зареєструйте $P_i P_j$ для кожної підключеної пари одиниць і середнє значення для всіх даних у міні-пакеті.

1.2. Поняття нейромережі Хебба

Навчання методом Хебба - це один з найстаріших алгоритмів навчання, і він значною мірою базується на динаміці біологічних систем. Синапс між двома нейронами зміцнюється, коли нейрони з обох боків синапсу (вхідні та вихідні) мають сильно корельовані виходи. По суті, коли вхідний нейрон

запускається, якщо це часто призводить до випалу вихідного нейрона, синапс зміцнюється. Слідуючи аналогії з штучною системою, вага кран збільшується з високою кореляцією між двома послідовними нейронами.

З точки зору штучних нейронів та штучних нейронних мереж, принцип Хебба можна описати як спосіб визначення того, як змінювати ваги між модельними нейронами. Вага між двома нейронами збільшується, якщо два нейрони активуються одночасно, і зменшується, якщо вони активуються окремо. Вузли, які, як правило, є як позитивними, так і негативними одночасно, мають сильні позитивні ваги, тоді як ті, які, як правило, супротивні, мають сильні негативні ваги.

Неврофізіолог Дональд Хебб в 40-х рр. Затвердив принцип, який став дуже впливовий на нейрокомп'ютер. Вивчаючи зв'язок між нейронами, один нейрон кілька разів збуджував інший нейрон, поріг збудження пізніше зменшувалася, тобто зв'язок між ними був полегшений повторюваним способом збудження. Це означає, що повторне збудження знизило поріг, або, точніше, це ефект збудження першого нейрону був посилений.

Дональд Хебб був першим, хто вважав, що "ефективність" даного нейрону, що сприяє пострілу іншого, може збільшуватися, оскільки ця клітина неодноразово задіяна в активації другого (Хебб, 1949). Таким чином, основним принципом вивчення хеббів в нейронних мережах є те, що 'одиниці, що активуються разом, проводяться разом'. Ця ідея узгоджується з нервовими механізмами довгострокового потенціювання (LTP) (Bliss & Lomo, 1973) та довготривалою депресією (LTD) (наприклад, Artola, Brocher & Singer, 1990). LTP та LTD впливають на те, якою мірою активність у відправляючому нейрона призводить до деполяризації приймаючого нейрону, впливаючи на ефективність синапсів або перехрест між нейронами. ЛТП є тривалим потенціацією (збільшенням) синаптичної ефективності, тоді як ТД є тривалим депресією (ослабленням) синаптичної ефективності. LTP і LTD, ймовірно, покладаються на іони кальцію, що вводять постсинаптичні дендрити через канали NMDA. LTP може бути

зрозумілим в термінах одночасної пресинаптичної та постсинаптичної активації. Пресинаптична активація викликає вивільнення збуджуючого глутамату нейромедіатора, який потім може зв'язуватися з та відкрити постсинаптичні NMDA рецептори. Постсинаптична активація викликає іони магнію для виходу з відкриття каналів рецептора NMDA, які вони інакше блокуватимуть. Іони кальцію можуть потім вводити постсинаптичний нейрон, де вони беруть участь у серії молекулярних змін, що в кінцевому підсумку збільшує загальну синаптичну ефективність. Обидва пресинаптична та постсинаптична активація, таким чином, необхідні для цього процесу. Нейрони, що пожежа разом, можуть з'єднатись через ці збільшення синаптичної ефективності. LTD може виникнути, коли менша концентрація кальцію потрапляє в постсинаптичний нейрон (Artola et al., 1990; Bear & Malenka, 1994; Lisman, 1994). Цей процес може відображати відсутність узгодженої активності між пресинаптичними та постсинаптичними нейронами. Нейрони, які не спляють разом, не з'єднуються через це зниження синаптичної ефективності.

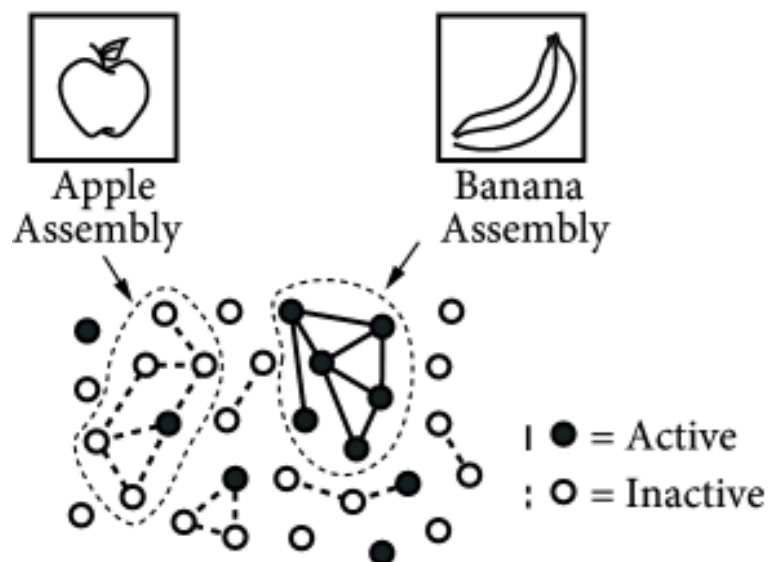


Рисунок 1.6 – Ілюстрація сильних та слабких поєднань

Екологічна обґрунтованість. Окрім того, що біологічно правдоподібно, механізми, що лежать в основі вивчення Хебба, є екологічно правильними. Ці механізми покладаються тільки на вхідні дані, що надходять в систему для вироблення моделей діяльності. Навчання протікає на основі цих

результуючих моделей діяльності. Таким чином, такі моделі можуть навчатися без необхідності виробляти конкретні відповіді, без будь-яких явних завдань або спрямованих спроб навчання, а також не вимагаючи викладання сигналів. Мережева реалізація хеббейського вивчення гебінського навчання здійснюється в моделях нейронних мереж через зміни сили тяги підключення між одиницями. Сила ваги з'єднання визначає ефективність передавальної одиниці при активації приймаючої одиниці і, таким чином, фіксує цей аспект синаптичної ефективності. У процесі вивчення геббейки ваги змінюються як функція рівнів активності одиниць. Завдяки такому навчанню ваги намагаються відображати статистичні закономірності в навколишньому середовищі, а мережі самоорганізуються, щоб різні навчальні заклади навчали представляти різні екологічні закономірності.

Багато різних типів мереж використовують вивчення геббейського стилю (наприклад, Bienenstock, Cooper & Munro, 1982; Kohonen, 1984; Miller, 1994). У мережі Kohonen (Kohonen, 1984) мережа представлена певним стимулом, і активізується блок "переможець" (той, що має найбільше вхідних даних) разом із сусідніми одиницями, причому активація зменшується з відстані від переможця. Така схема активації імітує кластери діяльності, що виникають в системах, коли зв'язки локально збуджують, але глобально гальмують (Konen & Von Der Malsburg, 1994). Зі вченням Хеб'єйна в такій мережі вага змінюється як функція одночасної діяльності підрозділів; тому що сусідні одиниці активізуються разом, їхні ваги змінюються аналогічними способами, такими, що одиниці навчаються реагувати на аналогічні матеріали як їх сусіди. Це дає топографічні карти, подібні до тих, що знаходяться в мозку, де групи сусідніх нейронів відповідають подібним чином до певних подразників (наприклад, Blasdel & Salama, 1986; Livingstone & Hubel, 1988; Merzenich & Kaas, 1980). Як викладено нижче, мережі Kohonen також використовувались для вивчення ефектів критичного періоду в сприйнятті мови.

Багато особливостей вивчення робіт Хебба є актуальними для теорії та даних розвитку, таких як здатність витягувати статистичні закономірності з навколишнього середовища. Діти дуже добре вивчають такі закономірності від порівняно короткого впливу слухових чи візуальних стимулів (Aslin, Saffran & Newport, 1998; Kirkham, Slemmer & Johnson, 2002; Maye, Werker & Gerken, 2002; Saffran, Johnson, Aslin & Newport, 1999 p.). Ці статистичні здібності для навчання є загальними для дорослих (Fiser & Aslin, 2001; Saffran et al., 1999) та інших видів (Hauser, Newport & Aslin, 2001). Хеббейське навчання може підтримувати таке статистичне навчання. Наприклад, коли мережі представлені зображеннями натурних сцен, вивчення хеббейків дозволяє їм створювати уявлення, які фіксують важливі статистичні співвідношення, присутні у зображеннях, а саме регулярне присутність країв у зображеннях (наприклад, Blair et al., 1998; O'Reilly & Munakata, 2000). Отримані уявлення включають багато важливих властивостей кортикальних рецептивних полів, таких як детектори краю, що варіюються за розміром, положенням та орієнтацією. Хеббейське навчання також дозволяє моделям отримувати статистичні закономірності мовних входів, наприклад, ступінь, в якій слова співпадають у середовищі (O'Reilly & Munakata, 2000). Ця інформація про спільне вживання є надзвичайно корисною для захоплення семантичної інформації про слова (Landauer & Dumais, 1997). Зауважте, що статистичне навчання не є унікальним для хеббейських моделей. Це загальна здатність, що досліджується в різних моделях нейронних мереж (наприклад, Altmann & Dienes, 1999; Christiansen & Curtin, 1999; Lee & Seung, 1999; Rogers & McClelland in press; Seidenberg & Elman, 1999).

Unsupervised Hebb Rule

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

input
actual response

Vector Form:

$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha \mathbf{a}(q) \mathbf{p}^T(q)$$

Training Sequence:

$$\mathbf{p}(1), \mathbf{p}(2), \dots, \mathbf{p}(Q)$$

Рисунок 1.7 – Правило навчання Хебба

Однак вивчення Хебба може бути особливо важливим для певних видів статистичного навчання (а також інших видів навчання), які, як видається, трапляються випадково, без явного завдання або спрямованої спроби навчання. Отже, інша важлива риса геббейського навчання для розвитку - це його автоматичний, самоорганізуючий характер. Цей вид навчання може проходити просто у відповідь на масив вхідних даних з навколишнього середовища, без урахування того, які результати повинні бути отримані у відповідь на ці входи. Навпаки, в алгоритмах, керованих помилками, такими як backprop, навчання вимагає, щоб відповіді мережі були порівняні з цільовими відповідями. Такі алгоритми, керовані помилками, можуть бути переосмислені з точки зору мереж, які виробляють результати, які передбачають, що відбудеться далі в середовищі, з наступними введеннями, що надають цільову інформацію (McClelland, 1994; Munakata, McClelland, Johnson & Siegler, 1997); це унеможливорює вироблення результатів та їх порівняння з явним тренувальним сигналом. Це переосмислення алгоритмів, керованих помилками, може розширити їх відповідність до цілого ряду областей розвитку. Тим не менше, вивчення геббейського навчання може забезпечувати тісніший зв'язок із типами навчання, що показуються немовлятами в певних випадках, наприклад, статистичних завдань навчання. Наприклад, ніж діти,

що формують прогнози та вивчають їх помилки, що базуються на коротких впливів на довільні та швидкі потоки фонем або візуальних послідовностей, може бути більш правдоподібним, що механізми навчання хебівського типу автоматично витягують статистичні закономірності з таких подразників.

Чому у розвитку розвиваються критичні періоди (або чутливі періоди), коли система виглядає набагато більш чутливою до впливу на навколишнє середовище, ніж на пізніші віки? Наприклад, люди, як правило, набагато краще вивчають мову (і зокрема, синтаксис) до статевого дозрівання, ніж після. Крім того, немовлята вивчають фонemi, якими вони піддаються, протягом перших місяців життя, тоді як старші діти та дорослі мають складніші завдання. Такі критичні періоди не обмежуються мовою, ані людьми. Наприклад, вони також спостерігаються в ефектах візуальної депривації у кішок, макаки та людей, а також у відбитковому поведінці курчат. Хеббові механізми можуть відігравати важливу роль у таких критичних періодах. Оскільки зв'язки змінюються для посилення певних нейронних реакцій на подразники, це може зменшити можливість інших відповідей, що може послабити навчання. Тут ми зосереджуємось на моделі, яка демонструє, як такі механізми можуть призвести до створення "молодих" мереж для розробки відповідних уявлень про фонemi у навколишньому середовищі, тоді як "старі" мережі не можуть це зробити. Ця модель використовує варіант мережі Кохонена та нормований алгоритм навчання Хебба. Мережі були представлені з фонем, представленими як закономірності діяльності на вхідному шарі. Деякі зразки були добре відокремлені, що відповідає легко виділеним фонемам. Інші шаблони перекриваються, що відповідає важким для розрізнення фонем, таких як / l / та / r /. І інші шаблони були змішаними версіями перекриваючих візерунків, що відповідають одиночним фонем, як японська фонема, що включає звуки / l / та / r /. Одна група мереж була "піднята" в англomовному середовищі, з добре розділеними та перекриваючими моделями. Друга група мереж була піднята в японській мові, з добре відокремленими та змішаними візерунками,

а пізніше була поміщена в середовищі англійської мови. Мережі продемонстрували критичний період навчання, щоб відрізнити накладені входні шаблони (наприклад, / r / i / l /). Молоді моделі представляли такі входи як різні шаблони у представницькому шарі, тоді як старші моделі натомість представляли ці входи як єдиний шаблон у представницькому шарі. Це відповідає молодим слухачам англійської мови / r / i / l / як окремі звуки, тоді як старші викладачі англійської мови (зокрема ті, хто вивчав японську мову як свою рідну мову) не можуть чути різницю між / r / i / l / звуками. Як молоді мережі навчилися правильно представляти фони у своїх середовищах, і чому старші мережі не змогли це зробити? Різні входи мали тенденцію активувати різні комбінації перцептивних одиниць. Навчання хеббів посилювали зв'язки між одиницями введення та сприймальними одиницями, які вони активували, провідних мереж належним чином представляти фонем у їх "рідних" середовищах. Модель, використовувана для вивчення критичних періодів у сприйнятті фонем. Для мереж, що виховуються з єдиним змішаним звуком, вивчення хеббів призвело до міцних зв'язків з відповідними одиницями введення до єдиного перцептивного представлення. Коли ці мережі потім були піддані окремим, але збігаючим, / l / та / r / звукам, ці входи мали тенденцію активізувати одиничне перцептуальне представлення. Навчання хеббів служило підтримці такої тенденції представляти окремі входи як єдиний звук, шляхом зміцнення зв'язків між одиницями вводу, що представляють різні звуки, та спільним представницьким представницьким представленням. В результаті, навіть при розширеному навчанні на окремих / l / and / r / звуках, мережі, які вперше дізналися єдиний змішаний звук, не могли активувати різні чіткі звуки.

Таким чином, розглянута схема Хебба для критичних періодів запропонувала спосіб покращити вивчення співвіднесених фонем після критичного періоду. Перебільшені / l / and / r / звуки можуть викликати чіткі уявлення про перцепцію. Навчання Хебба могло б зміцнити зв'язки, що

підтримують такі чіткі уявлення, збільшуючи ймовірність того, що / l / and / r / inputs активізують відмінні перцептивні подання. Ступінь перебільшення / l / i / r / звуків може потім зменшуватися поступово, зберігаючи чіткі уявлення для двох входів, доки вони не потраплять до природного діапазону. Навпаки, просте багаторазове враження від дублюючих подразників не повинно покращити продуктивність, оскільки вивчення хебьєнів лише зміцнить схильність почути ці фони як єдиний звук. Ці прогнози з моделі хебніків про те, як підвищити ефективність, були підтверджені у дорослих, котрі навчалися в Японії.

2. ДОСЛІДЖЕННЯ ПОБУДОВИ НЕЙРОМЕРЕЖ

2.1. Математична модель нейромережі

Зорові образи - основа сприйняття людиною навколишнього світу. Важко знайти область діяльності, яку в тій чи іншій мірі не зачіпала б інтелектуальна обробка цифрових зображень [11]. Прикладами практичних завдань є: розпізнавання рухомих об'єктів (транспортних засобів, літальних апаратів), комп'ютерні технології в сфері транспорту, екологічні та метеорологічні проблеми, медичні програми на основі відеоінформації, системи відеоспостереження та охорони, обробка супутникових та аерозображень, радіолокаційні зображення поверхонь і об'єктів, аналіз біометричних показників (відбитки, підписи, особи, зіниці і т.д.), системи неруйнівного контролю обладнання, робототехнічні системи, Системи спеціального призначення і багато інших. Конкретні завдання - аналіз економічної інформації, розпізнавання ієрогліфів, номерів автомобілів і залізничних вагонів з метою контролю проходження вантажів і транспортних потоків, автоматичний аналіз і розуміння документів (включаючи підписи), розпізнавання і візуалізація біомедичних сигналів, аналіз мікротріщин в корпусах морських суден, пристрої для сортування риби та візуального аналізу берегової лінії в морських регіонах, космічний моніторинг водних екосистем, системи відновлення і розпізнавання зображень в криміналістиці, томографія та аналіз рентгенограм, системи розуміння сцен, захисту рухів роботів від зіткнень, розпізнавання зображень друкованих плат при проектуванні, контроль процесів горіння і багато інших [11].

Існуючі методи вирішення проблеми аналізу об'єктів в реальних візуальних сценах не дозволяють в достатній мірі вирішити багато важливих завдань, тому що модель їх функціонування принципово не враховує впливу зовнішніх помилкових впливів і виникають через це часткових уявлень об'єктів. Необхідна розробка більш складних структурно-ієрархічних

підходів, заснованих на аналізі сукупності частин об'єктів. На мій погляд, основними ідеями, які можуть бути покладені в основу побудови і ефективного застосування таких систем є: групова модель перетворень і теорія нормалізації [1-3, 61], кореляційний підхід [7-16], теорія грануляції даних [10, 13], інтегральні і диференціальні методи аналізу багатовимірних сигналів [8, 12, 16], інваріантні методи формування локальних ознак (ЛПО) і обробка сигналів гауссовськими фільтрами [5, 8], інтелектуальні методи структурного аналізу і зіставлення [6], перетворення Хафа [9].

Весь спектр методів аналізу зображень в інтелектуальних системах традиційно поділяється відповідно до етапом обробки зображення на три групи: попередня обробка (фільтрація, сегментація, виділення контурів, характерних точок та ін.); побудова описів (обчислення векторів або множин ознак, формування структурних елементів та моделей); інтерпретація описів (прийняття рішення про відповідність сформованих в процесі аналізу моделей і еталонних описів, оцінка параметрів об'єктів). Останні два етапи - побудови і інтерпретації описів в даний час відносять до проблемних завдань штучного інтелекту.

Подібні етапи обробки відбуваються і в природних системах, наприклад в людському оці. Поряд з цим у людського зору існує інше, так зване синтетичне сприйняття, коли спостережуваний об'єкт сприймається відразу як цілісний образ. При синтетичному сприйнятті впізнавання відбувається рефлекторно, без докладного аналізу всього інформаційного поля зображення, часто лише за окремими фрагментами об'єкта. Наприклад, фахівцями з окремих деталей на основі візуального огляду впевнено розпізнаються численні марки автомобілів. За фрагментами, лише частково представляє розпізнається об'єкт, людський зір відновлює весь цілісний образ об'єкта, спираючись на апріорно задану модель. Здатність подумки добудовувати образ з уривчастих елементів - фундаментальна властивість зорового «апарату» людини.

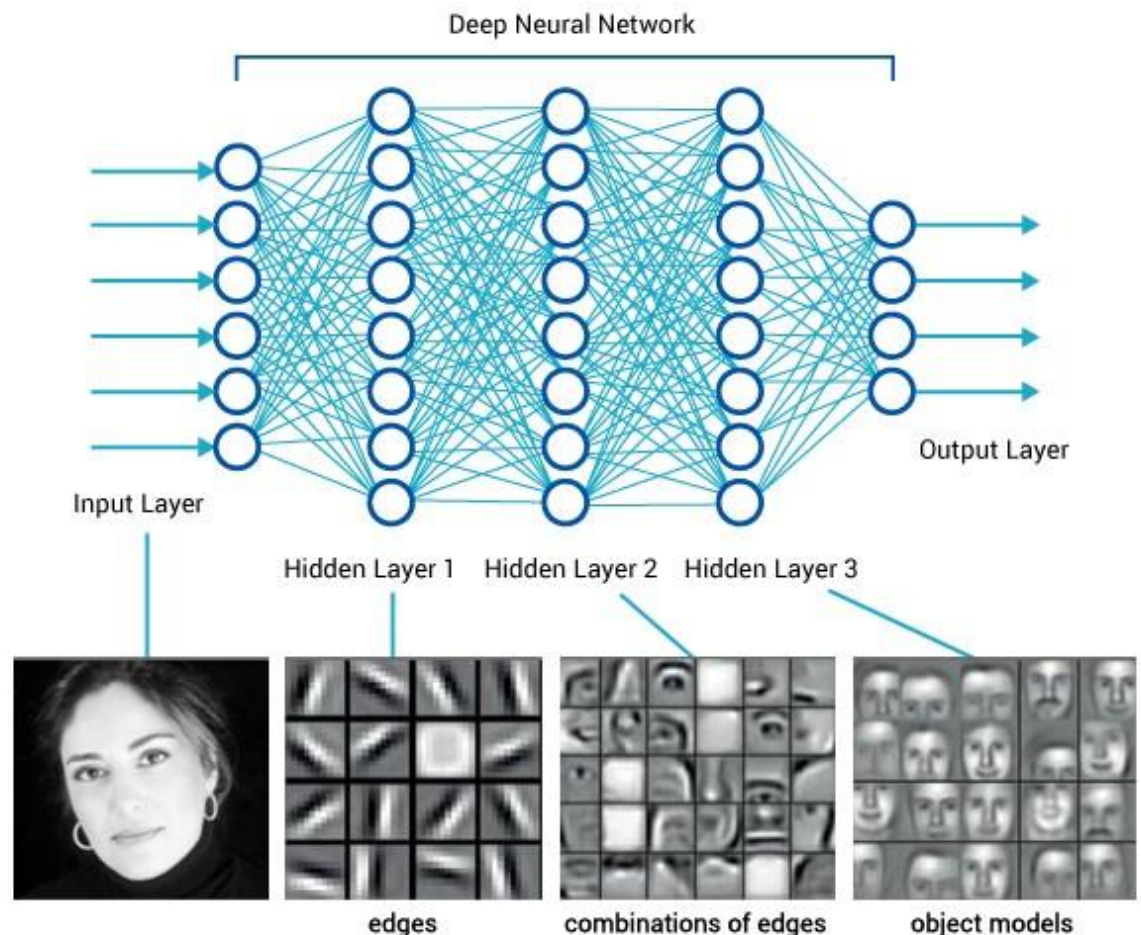


Рисунок 2.1 – Приклад розпізнавання обличчя за фрагментами

У біологічному сприйнятті, доповнюючи один одного, ефективно використовуються обидві моделі зображення - як повна (аналітична), коли аналізується все безліч точок зображення, так і часткова (синтетична). Існуючі ж штучні побудови процесів сприйняття в СТЗ спираються, в основному, на аналітичну модель, універсально підходить для всієї різноманітності візуальних об'єктів, представлених функцією яскравості в деякому полі зору.

Відмінні риси інтелектуальних СТЗ: вміння виділяти суттєву інформацію з наявного безлічі ознак; здатність до навчання і узагальнення знань з метою застосування їх в нових ситуаціях [26]; відновлення та аналіз подій за неповної інформації про них; здатність визначати цілі і формулювати плани дій системи для досягнення цих цілей. Ухвалення

рішення при неповній інформації - одна з основних характеристик інтелектуальних систем.

Дослідники схилиються до думки, що підвищення ефективності методів аналізу зображень при вирішенні практичних завдань можна досягти тільки шляхом ретельного вивчення процесів формування зображень, тобто максимально точно враховувати моделі шуканих об'єктів і перешкод. Це дозволяє поліпшувати якість роботи систем і забезпечує можливість розробляти оптимальні схеми прийняття рішень. Реальність придбали системи, в основу яких покладено досить велика інтелектуальна база знань. Такі системи на кожному кроці твердо «знають», що вони розпізнають або аналізують.

У даній роботі основні теоретичні положення і дослідження стосуються двовимірних зображень. функція і інформація, що зберігається в ній у вигляді двовимірних візуальних об'єктів, є основою всіх обговорюваних методів і моделей. Структури даних в термінах представлення тривимірних об'єктів не розглядаються, не дивлячись на те, що природа виникнення реальних зображень, а також ефектів появи тіней, фону і багатьох візуальних об'єктів за своєю суттю носить тривимірний характер. Найбільших успіхів теорія і практика тривимірної візуальної обробки досягла в сучасній комп'ютерній графіці [12]. Користуючись загальними структурами даних і застосовуваними математичними моделями, теорія розпізнавання зображень (побудова і зіставлення описів по зображеннях) і комп'ютерна графіка (синтез зображень по їх описами) розвиваються спільно, збагачуючи один одного новими ідеями і програмно-технічними розробками.

Очевидно, що ми повинні оновити ваги в нашій нейронній мережі, щоб дати правильний вихід на вихідному шарі. Це є основою навчання нейронної мережі. Ми будемо використовувати зворотне поширення для цих оновлення ваги. Це просто означає, що вхід подає, помилки підраховуються та відфільтровуються, хоча мережа вносить зміни до ваг, щоб спробувати зменшити помилку.

Зміни ваги розраховуються за допомогою методу спуску градієнта. Це означає, що ми дотримуємося найжорсткішого шляху до функції помилки, щоб спробувати минути його. Якщо різниця є позитивною, ми повинні рухати градієнт функції активації, і якщо її негативна, ми повинні рухатися вниз по градієнтові функції активації.

Гарним інтуїтивним способом думати про нейронну мережу є думка про те, що намагається зробити модель лінійної регресії. Лінійна регресія вимагатиме деяких входів і виведе лінійну модель, яка приймає кожне значення входних значень, деякі оптимальні коефіцієнти зважування моделей, і намагається відобразити суму цих результатів до вихідної відповіді, яка тісно відповідає дійсному виходу. Коефіцієнти визначаються шляхом знаходження значень, які мінімізують деяку метрику помилок між бажаним вихідним значенням та значенням, яке вивчається моделлю. Інший спосіб сказати, що лінійна модель спробує створити коефіцієнт множників для кожного вводу і сумувати всі з них, щоб спробувати визначити зв'язок між (декількох) входними та (зазвичай, єдиними) вихідними значеннями. Цю ж модель можна майже сприймати як основний будівельний блок нейронної мережі; єдиний блок персептрон.

Але одиничний блок персептрона має ще одну частину, яка буде обробляти суму зважених даних нелінійно. Для цього це зазвичай використовує функцію розбиття (сигмоїд або танх). Таким чином, у вас є основна одиниця прихованого шару, який являє собою блок, який збирає сукупність зважених входних даних - він потім передає і підсумовує відповідь на нелінійну функцію, щоб створити відповідь вихідного вузла (прихованого шару). Упереджений блок подібний до лінійної регресії, постійне зміщення, яке додається до кожного оброблюваного вузла. Через нелінійний блок обробки ви більше не обмежуєтесь лише лінійними відповідями (як у моделі лінійної регресії).

Гаразд, але коли є багато одиниць персептрону, які працюють разом, кожен може мати різні множники ваги вхідного сигналу та різні відповіді (навіть якщо ВСЕ обробляє один і той же набір входів з тим самим нелінійним блоком, описаним раніше). Те, що робить різні відповіді, полягає в тому, що кожен з них має різні коефіцієнти ваги, які вивчаються нейронною мережею через тренування (деякі форми включають градієнтний спуск). Результат всіх персептронів обробляють знову і передаються на вихідний шар, так само, як окремі блоки обробляються. Тоді питання полягає в тому, як визначаються правильні ваги для всіх блоків?

Найпоширенішим способом вивчення правильної ваги є початок випадкових ваг і вимірювання відповіді помилки між дійсним фактичним виходом і результатом вивченої моделі. Помилка, як правило, проходить назад через мережу, і алгоритм зворотного зв'язку індивідуально збільшить або зменшує ці ваги, частково з помилкою. Мережа буде багаторазово повторювати, проходячи вперед, вимірюючи вихідну відповідь, а потім оновлюючи (пропускаючи зворотні коригування ваги) та коригуючи ваги, доки не досягне певного рівня помилки. У цьому пункті ви маєте модель регресії, яка може бути більш гнучкою, ніж модель лінійної регресії, це те, що зазвичай називають аппроксиматором універсальної функції.

В процесі функціонування мережі Хопфілда можна виділити два режими: навчання і класифікації. У режимі навчання на основі відомих векторів підбираються вагові коефіцієнти мережі. У режимі класифікації при фіксованих значеннях ваг і введення конкретного початкового стану нейронів виникає перехідний процес виду (2.1), що завершується в одному з локальних мінімумів, для якого $y(k) = y(k-1)$.

Навчання мережі Хопфілда за правилом Хебба

Для одного навчального вектора значення ваг можуть бути обчислені за правилом Хебба

$$w_{ij} = \frac{1}{N} x_i x_j ,$$

оскільки тоді $\frac{1}{N} \left(\sum_{j=1}^N x_i x_j x_j \right) = x_i$ (внаслідок біполярних значень елементів

вектора завжди $x_j^2 = (\pm 1)^2 = 1$).

При введенні більшої кількості навчальних векторів $x(k), k = 1, 2, \dots, p$ ваги w_{ij} підбираються згідно узагальненому правилом Хебба

$$w_{ij} = \frac{1}{N} \sum_{k=1}^p x_i^{(k)} x_j^{(k)}.$$

Важливим параметром асоціативної пам'яті є її ємність. Під ємністю розуміється максимальне число запам'ятати образів, які класифікуються з допустимою похибкою. Показано, що при використанні для навчання правила Хебба і при (1% компонентів образу відрізняється від нормального стану) максимальна ємність пам'яті складе всього лише близько 13,8% від кількості нейронів, що утворюють асоціативний пам'ять.

2.2. Дослідження складових та етапів побудови нейромережі

Створення нейронної мережі означає створення єдиної системи розумової свідомості, навченої для вирішення однієї проблеми або, щонайбільше, пов'язаних питань. Вона отримує вхідні дані, які вона нарізає і кубиків через мережу штучних нейронів, з надією класифікувати її, розглядаючи характеристики кожної частини та порівнюючи їх з відомими ресурсами.

Приклади нейронних мереж. Як вже згадувалося на початку, нейронні мережі мають безліч застосувань у широкому діапазоні галузей, від охорони здоров'я до військової оборони або управління ризиками. Давайте переглянемо список найцікавіших поточних застосувань.

Торгові платформи. Якщо правильні моделі прогнозування були використані до 2008 року, є шанс, що фінансова криза ніколи не станеться або не матимуть такої масштабної ситуації. Це лише питання часу, перш ніж брокери замінюються машинами. Ця тенденція продиктована високочастотною торгівлею, яка працює з низькими значеннями, але

величезними обсягами. Оскільки людям неможливо приймати правильні інвестиційні рішення кожні 15-20 секунд, це чудовий приклад для AI, який працює на нейронних мережах.

Рекомендаційні двигуни. Маркетинг розвивався з масового звернення до індивідуальної адаптації. Що більш особистий ви можете отримати з вашими пропозиціями, повідомленнями та рекомендаціями, тим більше шансів, що ви будете продавати та продавати багато. Нейронні мережі можуть дізнатись про переваги, а потім створювати подібні або пов'язані продукти рекомендації. Завдяки пов'язаним технологіям, таким як розпізнавання зображень, система може бути настільки витонченою, що вона може отримати зображення нарядів зірки як вхід і повернути список магазинів, де подібні предмети доступні для продажу.

Охорона здоров'я. Успіх виявлення раку нейронними мережами вже є прийнятним проривом. Це може виявити різні види злоякісних утворень, перш ніж підготовлені лікарі могли б запропонувати той же діагноз. Але це не єдиний спосіб AI може поліпшити наше здоров'я. Ще менш небезпечними для життя проблемами, такими як прогнозування менструального циклу, можуть скористатися розпізнавання образів, зроблені штучними нейронами, пропонуючи більш точний календар для людей, які прагнуть стати батьками. Ще одне використання передбачає прогнозування днів госпіталізації.

Обробка природних мов. Це одна з високих частот нейронних мереж, оскільки вона може бути використана для створення безпечного комунікаційного мосту між людиною та машиною. Коли NLP повністю функціонує, лише тоді користувачі зможуть задовільно замінити агенти колл-центрів та інші робочі місця, що відкриваються. Це буде дуже корисним для інструкцій з експлуатації, засобів усунення неполадок або навіть навчань.

Виходячи з однієї ідеї звичок, виявлення шахрайства на основі штучних нейронів викликає червоні прапори кожного разу, коли відбувається незвичайна транзакція. Ви, можливо, вже знайомі з перевіркою 'чи це справді ти?', Коли ви отримуєте доступ до електронної пошти з іншого пристрою. Це

те ж саме, коли ви вперше використовуєте свою кредитну картку за кордоном, або для великої покупки, система побачить це як незвичайне.

Від самостійних машин до безпечного світу. Це лише деякі з способів використання нейронних мереж. Звичайно, автономний автомобіль залишатиметься одним з головних додатків та мрією шанувальників AI. Розвиток такого проекту є не тільки важливим завдяки своєму кінцевому результату - автономному автомобілю, який з наміром призведе до того, що пробки стануть застарілими, однак проміжні розробки з метою отримання всіх систем. Ці посередницькі етапи дослідження допоможуть підвищити безпеку в інших областях, таких як аеропорти, школи та лікарні.

Управління ризиками та виявлення шахрайства. Люди, як звички, мають тенденцію повторювати свою поведінку. Зміна способів дій ми вимагаємо величезної кількості сили волі, спрямованості та рішучості. Тим не менш, в більшості випадків ми повертаємося до старих звичок. Це психологічне спостереження є основою використання нейронних мереж для оцінки ризику позик та іпотечних кредитів. Хоча показники FICO можуть бути покращені через рік або близько того, розглядаючи шаблони, що стоять за цими цифрами, можуть виявити, якщо наміри хороші і управління грошовими коштами є надійним, або якщо це просто справа обману системи.

Програми є нескінченними, від вибору потрібних кандидатів на роботу, до оцінки ризиків аварій та навіть прогнозування кліматичних змін. Майже в будь-якому секторі діяльності є проблема, яка може бути вирішена нейронними мережами. Тому має сенс взяти більш детальний огляд.

Побудова нейронної мережі. Процес трохи схоже на виготовлення тортів. Ви збираєте інгредієнти (штучні нейрони), змішуєте їх у чаші (створіть перший шар, який ви можете побачити), а потім покладіть все в духовку (приховані шари) і сподівайтесь на найкращий результат (вихідний шар).

Щоб заглибитись трохи технічніше, пропоную пройти кроки, необхідні для створення функціональної нейронної мережі. Оскільки ніхто не хоче

винаходити колесо, ви починаєте з імпорту деяких корисних бібліотек і, звичайно ж, наборів даних.

Далі, оскільки в кожному з нейронів у першому шарі немає визначених ваг (значення), вони отримують призначені випадкові числа від 0 до 1. Після цього ви можете перейти до веселої частини фактичного навчання мережі. На етапі прямого розповсюдження ви ходите по мережі та порівнюєте результати, які ви отримуєте з тими, кого ви очікували, щоб побачити, як далеко ви були. Це по суті доводить, наскільки хороша модель. На етапі зворотного розповсюдження ви маєте на меті зробити мережу вчитися з її помилки, починаючи з кінця на увазі і працюючи через систему в зворотному порядку. Цей процес зворотного зв'язку дає нові, більш точні ваги для початкового шару штучних нейронів.

Повторюючи цикл назад і вперед сотні разів (це називаються епохи), прогноз робиться досить точно, що його подальше вдосконалення неможливо покращити. Коли це спостерігається, прийшов час зупинитися, бо мережа переобтяжується. Перенасичення означає, що вона приділяє велику увагу навчальному набору даних, що вона більше не може бачити загальну картину, і вона буде марна в інших наборах даних.

Досить ефективним напрямом є побудова інваріантів на базі перетворень Хемінга.

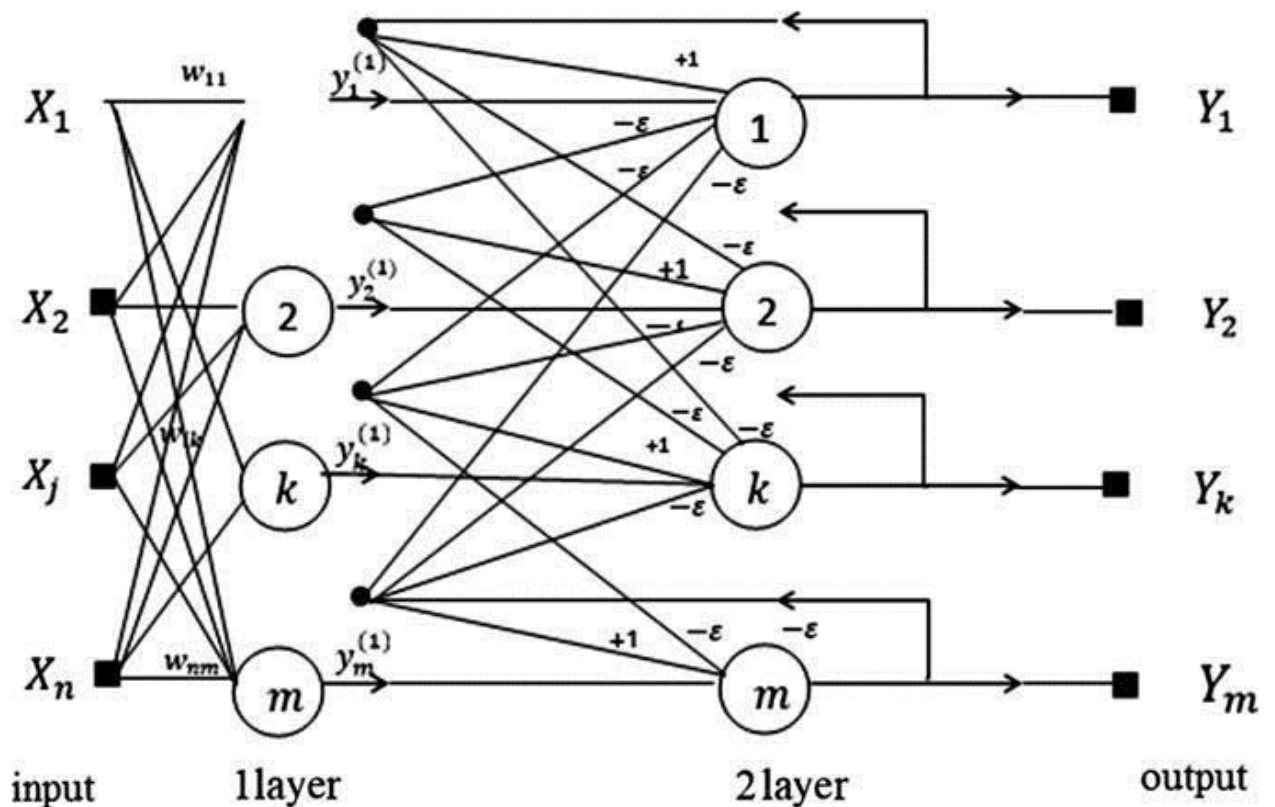


Рисунок 2.2 – Загальна схема нейронної мережі Хеммінга

Останнім часом в якості ефективного засобу для інваріантного опису зображень застосовують вейвлет-аналіз [28,]. По суті - це спектральний аналіз локальних характеристик зображення. У роботах [115, 168] розглянуті підходи до оцінки виду геометричних перетворень, знаходженню контурів і побудови інваріантів інтегрального типу. Пропонується в якості еталонної моделі ознак використовувати вектор коефіцієнтів одновимірного вейвлет-перетворення для кількох сфер зовнішньої та зсувів як параметрів [28], а також застосування вейвлетів для формування ХП [5]. Розглянуті методи мають ті ж переваги, що і спектральні підходи, додатково дозволяючи здійснювати локальний просторовий і енергетичний аналіз. До недоліків вейвлетного методів відносять великі витрати пам'яті для зберігання проміжних перетворень.

Окремим напрямком варто застосування систем диференційних ознак, в основі яких лежать похідні від функції яскравості. Найбільшого поширення ці підходи отримали в задачах аналізу контурів двовимірних об'єктів. В даний час диференціальні підходи застосовуються при формуванні

особливостей зображень на локальних ділянках [4, 5]. Математичне обґрунтування і особливості застосування цих методів розглядаються в роботі стосовно завданню обчислення ХП.

Фундаментальні положення теорії нормалізації зображень і застосування її в прикладних задачах вивчені і викладені у фундаментальній монографії Є.П. Путятіна і С.І. Аверіна [1], а також розвинені в наступних роботах [2, 3, 10].

Коротко суть цієї теорії полягає в наступному. Здається оператор нормалізації у вигляді $F(B)$, який відображає вихідне зображення B до ідеалу. Сенс переходу полягає в компенсації дії перетворення з групи G . При цьому спочатку обчислюють (оцінюють) величини параметрів, а потім здійснюють компенсацію дію, застосовуючи до функції зображення зворотне перетворення. Такі кроки завжди призводять до мети при груповій моделі перетворень, тому що кожен елемент g теоретично має протилежний g^{-1} . Надалі на безлічі еталонів проводиться пошук найбільш схожого еталона в сенсі деякої міри. На відміну від прізнакових методів, методи нормалізації здійснюють розпізнавання в просторі зображень, що дозволяє краще контролювати ситуацію і забезпечує хорошу перешкодозахищеність.

Розроблено велике розмаїття нормалізаторів і конкретних алгоритмів для їх реалізації [1-3]. Вони враховують види перетворень, включаючи метричні, афінні, проектні, використовують різні функціонали (і способи їх отримання) при побудові відображень в групі, будуються по паралельному або послідовному принципу, в цілому відрізняються хорошою помехозахищеністю і швидкістю. Крім того, в процесі нормалізації природним чином оцінюються параметри перетворень. Численні модельні експерименти і технічні застосування [1, 10] підтверджують ефективність нормалізації при вирішенні практичних завдань розпізнавання зображень об'єктів в СКЗ.

Кореляційні методи. Ці підходи набули поширення завдяки своїм оптимальним властивостям при дії перешкод [1, 8] і застосовуються в

різних просторах уявлення сигналів: зображення, спектри, ознаки, безлічі перетворень і т.д. Кореляційні принципи знайшли відображення в сучасних методах виділення і зіставлення множин ХП, тому що будь-який виділений ознака має відтінок оптимальності, а, значить, узгодженості з деяким ідеальним уявленням.

Для завдання розпізнавання зображень суть кореляційного підходу полягає в наступному. При відсутності перетворень кореляційний метод зводиться до пошуку аргументу оптимуму деякої міри близькості на безлічі еталонів відповідно до правила.

Це пояснюється об'ємними обчисленнями, викликаними необхідністю перебору всіх можливих дискретних значень параметрів для кожного з еталонів, а також формування перетворених еталонів. Незважаючи на це, кореляційний метод знаходить саме широке практичне використання в діючих кореляційно-екстремальних системах в силу високої перешкодозахищеності і оптимальності щодо дії поширених флуктуаційних перешкод адитивного типу. Найбільш детальний аналіз і систематизація кореляційних методів наведені в роботах [1, 8]. Ряд вдалих модифікацій цих методів націлені на підвищення швидкодії відповідних алгоритмів за рахунок часткового зниження надійності або скорочення області переборюючи [8]. Незважаючи на ці недоліки, в даний час кореляційний підхід залишається одним з основних практичних способів розпізнавання.

Застосування перетворення Хафа (Hough). Перетворення Хафа (ПХ) відносять до одного з найбільш потужних інструментів, що дозволяють здійснювати аналіз форм візуальних об'єктів, будувати алгоритми виявлення, оцінювання координат і розпізнавання [4, 9].

Зображення об'єкта й еталона представляються у вигляді множин СЕ. Це, як правило, точки, прямі, кола та інші елементарні подоб'єкти. Аналітично задається модель перетворення. Здається також деяка аналітична поверхня, де $x = (x_1, \dots, x_n)$ - вектор з простору сигналів, $a = (a_1, \dots, a_m)$ - вектор з простору параметрів перетворення. Поверхня, що задає зв'язок

сигналів і їх перетворень, включає в себе безпосередньо і вид перетворення. В результаті застосування в просторі сигналів виділяється підпростір, яке описується характеристичною функцією приналежності сигналу підпространству, де виконується умова. Область значень параметрів перетворень дискретизується з деяким кроком по кожній компоненті a_i вектора a . Формується m -мірний масив-накопичувач A , в який записуються значення спектральних відліків. Суть отримання спектрів полягає в тому, що для кожного вектора x знаходяться значення функції при всіх можливих значеннях параметра a . У масиві-накопичувачі A послідовно шляхом обробки кожного СЕ збільшуються на 1 (інкрементуються) значення тих осередків, для яких виконується. Тобто коли вектори параметрів знаходяться на одній поверхні. Після перегляду всіх точок вміст окремої комірки накопичувача відповідає величині спектрального відліку PX . У відповідності зі значенням максимуму в масиві A приймається рішення про величину параметра і про відповідність об'єкта еталону. Таким чином, критерієм прийняття рішення при використанні PX є число голосів, відданих за величину перетворення або клас об'єкта.

PX є одним з ефективних методів вилучення глобальної інформації з сукупності локальних властивостей об'єктів. Обчислювальні витрати цього методу значно зростають зі збільшенням числа і безлічі значень простору параметрів. Успіх в застосуванні PX досягнутий в практичних завданнях при розпізнаванні зображень японських ієрогліфів, при контурному аналізі форм хмар, об'єктів з формою кіл різних радіусів, об'єктів типу літаків і кораблів, а також в задачах спостереження за рухомими малорозмірними і протяжними об'єктами [28]. Деякі дослідники справедливо вважають методологічну близькість PX з визначенням кореляційного подібності. Зокрема, обчислення PX і функції взаємної кореляції для бінарних зображень дають практично ідентичні результати [17].

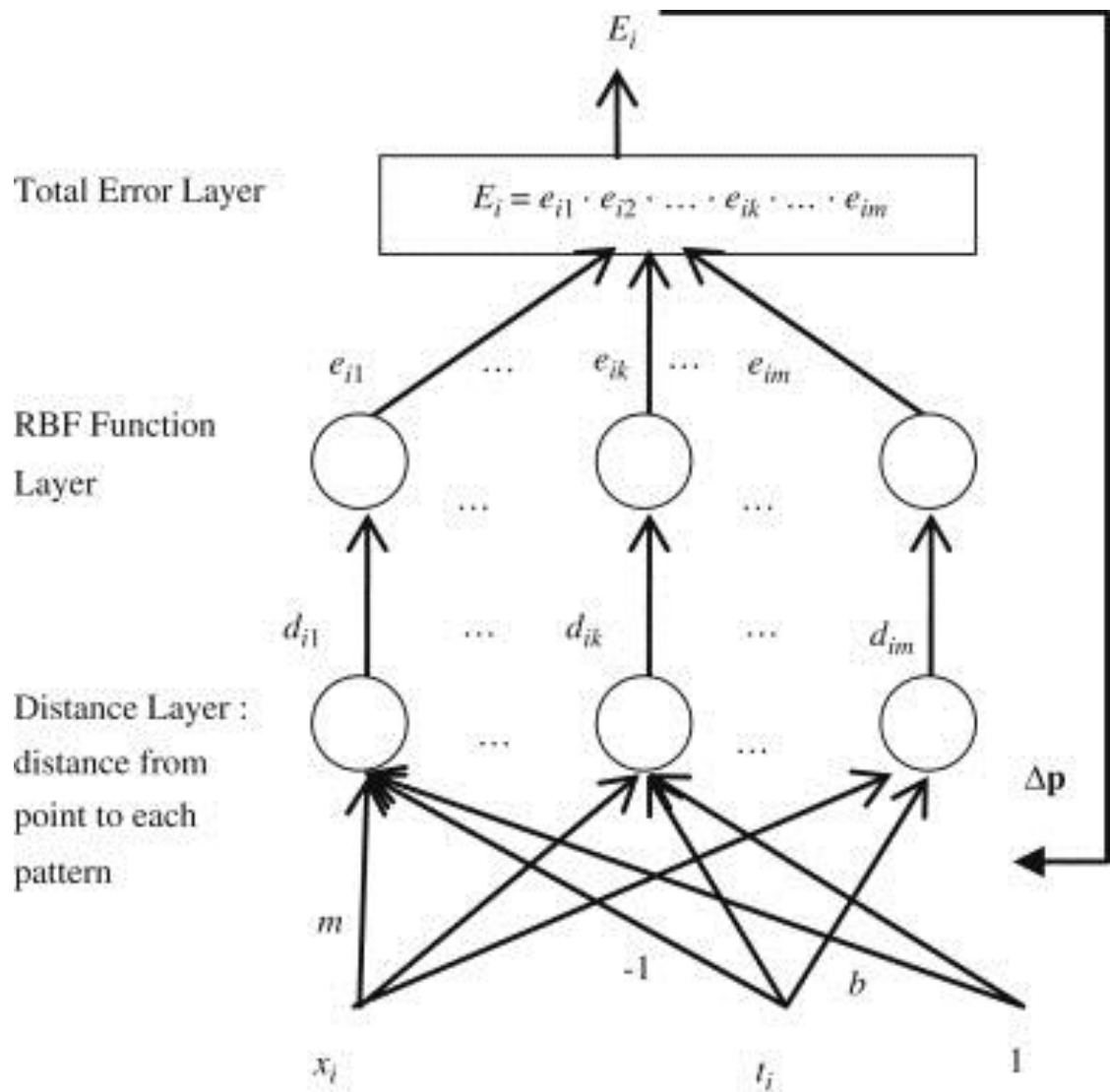


Рисунок 2.3 – Загальна схема перетворення Хаффа

Методи, які використовують ПХ, в обчислювальному плані дещо переважають над кореляційними методами, якщо вдається сформувати ефективне простір структурних ознак. У той же час і кореляційні методи допускають деяке посилення шляхом їх реалізації на безлічі ознак. У скільки разів звужується це простір в порівнянні з вихідним простором точок зображення, у стільки раз приблизно скорочується і час розпізнавання. Платою за це перевага є необхідність наявності додаткових обсягів пам'яті для зберігання значень параметрів в масиві-накопичувачі.

З інших поширених підходів до розпізнавання при дії перешкод слід відзначити групу статистичних методів [6-9], які часто безпосередньо пов'язані з розглянутими підходами. Справа в тому, що виступає критерієм і оптимізується на безлічі еталонів міра близькості описів може бути

представлена як значення статистичного критерію [9, 16], де вектори описів виступають в ролі вимірювань. Безпосереднє застосування статистичної теорії розпізнавання найбільш ефективно в разі наявних імовірнісних описів класів і об'єктів. Саме в цій ситуації можна здійснити оптимальне рішення [9, 16]. З огляду на, що в більшості завдань комп'ютерного зору через нескінченного розмаїття розпізнаваних об'єктів імовірнісний розподіл фрагментів і значень ознак важко уявити аналітично, класичне застосування статистичних критеріїв на практиці часто утруднено. З цих причин основним критерієм при розпізнаванні можна вважати значення заходи подібності описів [9,12].

Оскільки процес навчання відбувається практично миттєво, питання про слідкуючі властивості алгоритму настроювання не виникає. Ідея «нейрони в точках даних», що розвивається, у поєднанні з нечіткою логікою дозволяє вирішувати більш широкий клас задач в режимі послідовної обробки даних за умов нестаціонарності.

На основі узагальненої регресійної нейронної мережі (УРНМ), яка використовує концепцію «нейрони в точках даних» може бути реалізовано адаптивне F-перетворення, що є універсальним апроксиматором і ключова ідея якого полягає в нечіткому розбитті вихідного простору на множину нечітких підпросторів (кластерів), що перетинаються, і локальної апроксимації вихідної функції в цих підпросторах за допомогою засобів нечіткого виведення.

Процес навчання УРНМ включає в себе уточнення параметрів кластерів у формі:

$$\begin{cases} a_j(k) = a_j(k-1) + y(k), & j = 1, 2, \dots, h; \quad k = 1, 2, \dots, l, \\ b_j(k) = b_j(k-1) + 1, \\ c_j(k) = \frac{k-1}{k} c_j(k-1) + \frac{1}{k} x(k), & \text{якщо } |x(k) - c_j(k-1)| \leq r, \end{cases}$$

$$\begin{cases} a_j(k) = a_j(k-1), \\ b_j(k) = b_j(k-1), \\ c_j(k) = c_j(k-1), \end{cases} \quad \text{якщо} \quad |x(k) - c_j(k-1)| > r,$$

де h – кількість кластерів;

r – радіус впливу.

Така подібність дозволяє побудувати адаптивну процедуру обчислення дискретного F-перетворення, в якій компоненти F_j уточнюються послідовно у міру надходження даних з урахуванням можливої нестационарності функції $f(x)$. Такий підхід дозволяє пов'язати розташування вузлів F-перетворення з характером функції, що апроксимується, і змінювати їх кількість h залежно від необхідної точності.

2.4. Варіанти подолання недоліків нейронної мережі

Важлива в математичному плані проблема розпізнавання зображень по структурній сукупності фрагментів із застосуванням правила вище полягає в можливості ототожнення об'єкта і еталона не по повному фрагменту, а лише щодо відповідності його окремих частин, тобто по підмножині фрагментів. На основі визначення для цих підмножин може бути зроблений висновок про повну відповідність. Це в значній мірі залежить від моделі побудови. У строгому математичному сенсі дві функції можна вважати рівними між собою лише в тому випадку, коли їх значення співпадають на всій безлічі точок визначення [17]. Тому для обґрунтованого прийняття рішення за правилом (2.3) необхідно інше (часткове по безлічі визначення функцій) завдання подібності. Часткові подібності допускають ототожнення функцій по рівності на окремих ділянках. Ясно, що при використанні цієї моделі апіорно повинен бути заданий поріг для числа фрагментів або кількості співпадаючих точок. При цьому з суворой математичної точки зору потрібно розуміти те, що, ототожнюючи дві функції по їх однаковим частинам, насправді ототожнюються цілі класи функцій, у яких ці частини збігаються, а інші частини в той же час можуть відрізнятися.

При аналізі пошуку всіх слів зображень можна було б застосувати вірогідне трактування, наприклад, вважати, що зі збільшенням числа загальних фрагментів зростає ймовірність збігу. В такому випадку можна оперувати статистичними критеріями при виборі порога для прийняття рішення про еквівалентність. Для аналізу візуальних образів такий підхід не завжди доречний, тому що він передбачає відомим розподіл усіх опис і однакову важливість всіх фрагментів. Більш прийнятним для аналізу візуальної інформації, на нашу думку, є спосіб підрахунку числа еквівалентних (ідентичних в деякому плані) фрагментів або їх ознак. Поріг на ступінь еквівалентності при цьому задається апріорно і є ключовим параметром при прийнятті рішень про клас розпізнається об'єкта.

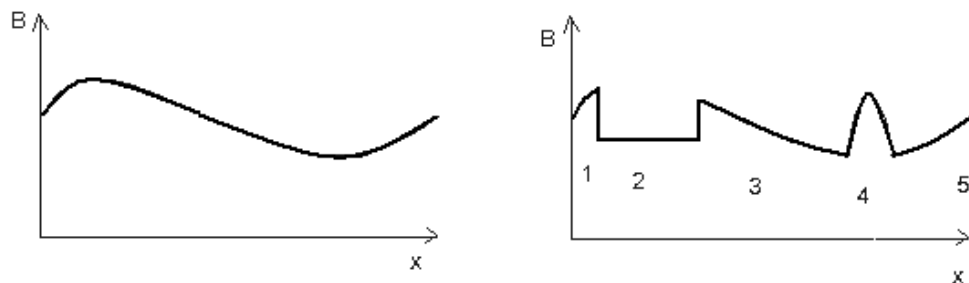


Рисунок 2.4 - Приклад сигналів, які збігаються на окремих ділянках:

1, 3, 5 -участкі збіги, 2, 4 - ділянки розбіжності

Таким чином, методи встановлення відповідності об'єктів по їх частин мають заздалегідь обумовлені обмеження. Наприклад, за допомогою них важко розпізнавати візуальні об'єкти, сильно схожі на великій кількості діляниць. До речі, такі об'єкти важко розпізнавати будь-яким з відомих методів. Однак в ряді практичних завдань ієрархічні методи можуть дати хороший результат в плані можливості прийняття рішення щодо часткового поданням цікавлять об'єктів [1, 5].

Поряд з поданням у вигляді множин ХП, при розпізнаванні візуальних об'єктів має сенс застосувати також компонентний уявлення (КУ), яке шляхом об'єднання ХП в групи формує більш складні структури - гранули, що складаються з підмножин ХП. Формування компонент здійснюється шляхом реалізації певної системи правил, наприклад, шляхом застосування відносин на безлічі ХП. Компонентне представлення різних типів і рівнів дозволяє здійснювати структурний аналіз візуальної інформації у вигляді

складових підоб'єктів. Основним завданням побудови і зіставлення компонент є надійне функціонування СКЗ в неповному просторі ознак і виключення помилкових компонент, що виникають через зовнішніх впливів. Розпізнавання в просторі ознак компонент вимагає розробки відповідних заходів подібності. Початкове перетворення Хеммінга при цьому можна розглядати або як нижній рівень ієрархії ознак, або як окремий випадок компонентного уявлення(КУ). На сформованому безлічі гранул застосовні всі формалізації, описані в попередньому розділі, включаючи побудову дескрипторів, функції відповідності та голосування. Нова якість компонентного уявлення полягає у включенні в процес вирішення структурних та групових зв'язків елементів з безлічі ХП, що в цілому підвищує достовірність аналізу за рахунок здійснення більш точної апроксимації форми об'єкта.

Концепція КП відповідає побудові стратифікованого опису візуальних об'єктів, одночасно володіє властивостями збереження цілісного уявлення і деталізацією опису, і що дозволяє аналізувати об'єкт на відповідному рівні абстрагування [7]. Для кожного рівня (страти) існують характерні особливості, закони і принципи, що відображають його властивості при розпізнаванні незалежно від інших рівнів. У той же час в рамках ієрархії опис кожного наступного рівня засноване на даних попереднього рівня. Загальний вигляд стратифікованого уявлення опису представлений на рис.5.1. При цьому аналіз і зіставлення описів може здійснюватися як незалежно по рівнях, так і спільно з використанням узагальненої заходи подібності, що включає безліч описів різних рівнів стратифікації. Елементи опису для окремих рівнів можуть мати як однаковий тип побудови, так і відображати особливості аналізованого рівня.

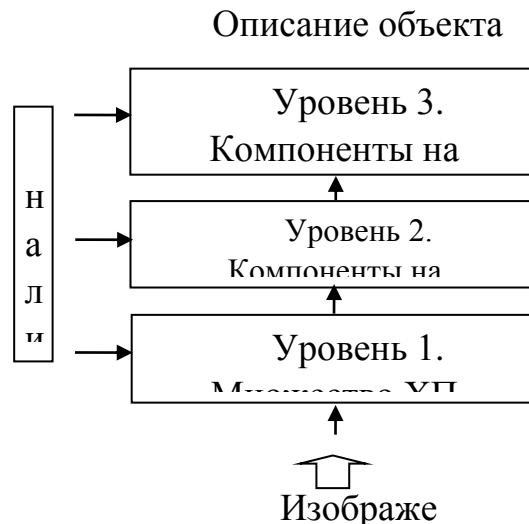


Рисунок 2.5 - Уявлення опису

Будемо розглядати розпізнається об'єкт як багатоеlementними сукупність (об'єднання) компонент, значення вектора ознак для компоненти по аналогії з ХП будемо називати дескриптором. Вибір способів обчислення дескрипторів і формування компонент в значній мірі визначає властивості класів об'єктів в плані їх поділу між собою [4, 45-49]. Компонента - це одиночний ХП або деяку підмножину ХП, сформований і розглядається як єдине ціле шляхом побудови відносин і має загальний дескриптор. Відносини дають можливість аналізувати поєднання ознак, які в інформаційному плані часто більш значущі для розпізнавання, ніж самі ознаки [20, 21]. В цьому аспекті не накладаються ніяких вимог на спосіб побудови компонент, зокрема, допускається їх просторове перетин. У той же час самі ХП характеризують властивості зображення в конкретних точках і незалежно між собою відображають особливості просторових утворень пікселів. З цієї причини ХП вже можна вважати компонентами аналізованого зображення. Компоненти відображають зв'язок ХП і можуть бути залежні. Рис. 2.7 ілюструє схему об'єктів.

Аналіз компонент більш високих рівнів в стратифікованому описі дозволяє більш строго врахувати внутрішню схожість структур об'єктів. У

відповідності зі значеннями заходів для подібності компонент формуються оцінки класів (рис.2.8).

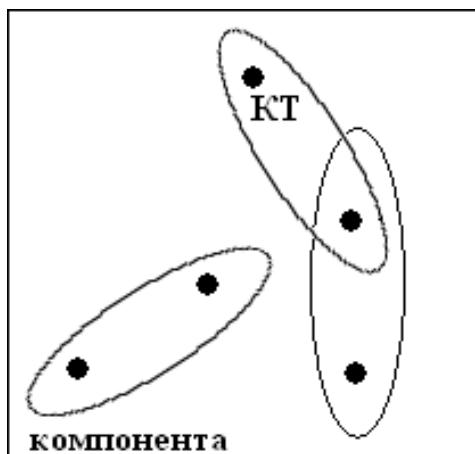


Рисунок 2.6 - Об'єкт у вигляді компонент

Етапи розробленого методу:

- 1) визначення безлічі ХП;
- 2) формування стратифікованого уявлення;
- 3) побудова заходи подібності з використанням ознак окремих страт і чи їх поєднань.

Використання відносин дає можливість на основі безлічі ХП отримати достовірні ознаки більш високого рівня [4, 12]. Відносини природним чином закладені в самій природі структурного уявлення як шлях отримання цілісної інформації про розпізнається об'єкті на основі опису у вигляді частин. В кінцевому рахунку, будь-який візуальний об'єкт можна формально описати шляхом побудови відносин на безлічі його елементів.

Ставлення (relation) - це абстрактне математичне поняття [12, 26].

Найбільшу популярність в аналізі зображень отримали топологічні відносини, які враховують взаємне розташування, а також метричні відносини, які оцінюють відстань між елементами. Такі топологічні відносини, як зв'язність, суміжність, інцидентність володіють стійкістю до поширених дефектів зображень, що призводить до порушення метричних відносин. Відстань як метричний ставлення в координатному розумінні інваріантної щодо поворотів і переносів, але не щодо масштабування.

Відносини розширюють безліч елементів, для яких можна встановити відповідність об'єктів.

Існує дійсно два рішення, які необхідно зробити щодо прихованих шарів: скільки прихованих шарів реально мати в нейронній мережі та скільки нейронів буде в кожному з цих шарів. Спочатку ми розглянемо, як визначити кількість прихованих шарів для використання з нейронною мережею.

Проблеми, що вимагають двох прихованих шарів, зустрічаються рідко. Проте нейронні мережі з двома прихованими шарами можуть представляти функції з будь-якою формою. В даний час немає теоретичних причин використовувати нейронні мережі з більш ніж двома прихованими шарами. Фактично, для багатьох практичних завдань немає причин використовувати більше одного прихованого шару.

Вибір кількості нейронів у прихованих шарах є дуже важливою частиною вирішення вашої загальної архітектури нейронної мережі. Хоча ці шари безпосередньо не взаємодіють із зовнішнім середовищем, вони мають величезний вплив на кінцевий результат. Слід уважно розглянути як кількість прихованих шарів, так і кількість нейронів у кожному з цих прихованих шарів.

Використання занадто багато нейронів у прихованих шарах може призвести до кількох проблем. По-перше, занадто багато нейронів у прихованих шарах може призвести до переобладнання. Надмірне накладення відбувається, коли нейронна мережа має стільки можливостей для обробки інформації, що обмеженого обсягу інформації, що міститься в навчальному наборі, недостатньо для навчання всіх нейронів у прихованих шарах. Друга проблема може статися навіть тоді, коли дані навчання достатні. Надзвичайно велика кількість нейронів у прихованих шарах може збільшити час, необхідний для тренування мережі. Кількість навчальних годин може збільшуватися до того моменту, коли неможливо адекватно тренувати нейронну мережу. Очевидно, що певний компроміс повинен бути досягнутий між надто багато і занадто мало нейронів у прихованих шарах.

Існує багато методів правильного визначення правильної кількості нейронів у прихованих шарах, таких як:

- Кількість прихованих нейронів має бути між розміром вхідного шару та розміром вихідного шару.
- Кількість прихованих нейронів повинно бути $\frac{2}{3}$ розміру вхідного шару, а також розмір вихідного шару.
- Кількість прихованих нейронів повинно бути менш ніж удвічі більшим за розмір вхідного шару.

Ці три правила є відправною точкою для розгляду. Зрештою, вибір архітектури для вашої нейронної мережі призведе до випробувань та помилок.

Для створення програми потрібно було підібрати доволно гнучке середовище. Вибір впав на програмне середовище Delphi та мову Pascal. Паскаль був створений видатним швейцарським комп'ютерним науковим співробітником Ніклаусом Віртом.

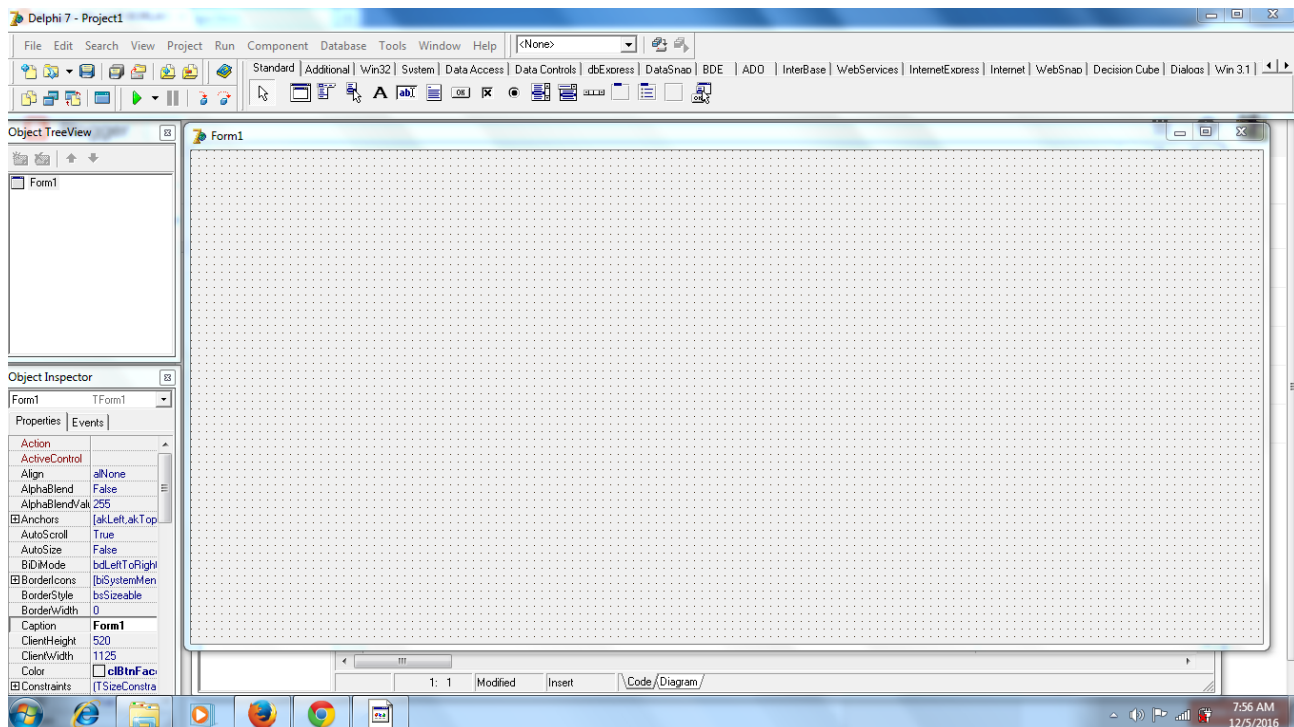


Рисунок 2.8 – зображення середи розробки

IDE або інтегроване середовище розробки - це інтерактивна програма, яка забезпечує всі основні функції, необхідні для створення, редагування, компіляції, посилань та налагодження програмного забезпечення. Перед

впровадженням IDE програмісти використовували окремі інструменти для кожного етапу процесу програмування. IDE дозволяють програмісту бути більш ефективними, забезпечуючи єдине послідовне розташування для виконання основних завдань, пов'язаних з кодуванням. Вони також допомагають програмістам уникати помилок, надаючи форматування синтаксису мови у редакторі. Хоча ви все ще можете виконати задачі програмування за допомогою окремих інструментів, переваги управління проектом IDE зазвичай перевищують зусилля, необхідні для того, щоб дізнатися, як працює IDE.

Сьогоднішні середовища інтелектуальної власності мають графічні інтерфейси користувача та мають мало спільного з Dartmouth BASIC - першою мовою програмування, яка використовувала IDE. Розроблений у 1964 році, Дартмут BASIC забезпечив командний інтерфейс, який включав в себе редагування, компіляцію та налагодження. Інші ранні IDE включають Maestro I та Softbench. На початку 80-х років компанія-програвач Borland представила Turbo Pascal, інтегрований набір програм для мови Pascal. Turbo Pascal не був достатньо потужним, щоб бути використаний для великомасштабних проектів програмування, але це був потужний прогрес над BASIC для початківців програмістів. За даними журналу Byte, за перші два роки після його випуску продано близько чверті мільйона примірників Turbo Pascal було розпродано.

У середині 90-х років Borland розвинув Turbo Pascal в інструмент "швидкого застосування" (RAD), який називається Delphi. Він був написаний в об'єктно-орієнтованій версії Паскаля під назвою Object Pascal і включав складну IDE. Протягом 90-х років Borland випускає все більш потужні версії Delphi, додаючи підтримку для динамічних масивів, перевантаження методу, взаємодії з базами даних та підтримку 32-розрядних операційних систем.

У 2008 році Borland продала відділ програмного забезпечення, відповідального за Delphi, Embarcadero Technologies. Embarcadero продовжує підтримувати Delphi, який наразі доступний у Microsoft Windows і створює

код, який працює на 32-розрядних та 64-розрядних ОС Windows, Mac та на мобільних пристроях Apple і Android. Якщо цікаво дізнатися, як програмувати за допомогою Delphi, вам потрібно мати доступ до комп'ютера під керуванням Windows.

Остання версія Embarcadero Delphi - це XE5. Головне його перевага - це можливість використовувати той самий код програми для різних платформ і пристроїв, включаючи ПК, ноутбуки, смартфони та планшети. Додаткові можливості, включаючи FireDAC, бібліотека, яка дозволяє додаткам підключатися до корпоративних баз даних, таких як MySQL, Oracle, DB2 та інші. Delphi XE5 IDE також підтримує візуальний розвиток, що дозволяє проектувати, створювати прототипи та реалізовувати додатки з використанням візуальних компонентів. VCL (Visual Component Library) - це візуальна структура, яка дозволяє миттєво змінювати зовнішній вигляд програми, застосовуючи заздалегідь визначені візуальні стилі до існуючого коду.

Lazarus - це конкуруюча RAD IDE, яка забезпечує майже таку ж функціональність, як Delphi. Його можна безкоштовно завантажити з веб-сайту "Лазар". Як і в Delphi, Lazarus надає крос-платформний IDE, який підтримує кілька платформ. Платформи, які підтримує Lazarus, включають Windows, Mac, Linux, Raspberry Pi, BSD та різні пристрої, включаючи Nintendo. Бібліотека класів Lazarus (LCL), яка нагадує Delphi VCL, підтримує графічний дизайнер з перетягуванням і падінням з більш ніж 200 компонентів, які полегшують створення графічних користувацьких інтерфейсів.

Lazarus IDE містить редактор коду, який забезпечує виділення синтаксису та завершення коду. Вона призначена для розробки коду Pascal за допомогою Free Pascal, компілятора з відкритим кодом для Object Pascal. Free Pascal сумісний з Delphi Pascal і підтримує багато однакових бібліотек. За даними веб-сайту "Лазар", "Free Pascal" достатньо обґрунтовано для

використання як студентами-програмістами, так і розробниками комерційних програмних продуктів.

3. РОЗРОБКА НЕЙРОМЕРЕЖІ ХЕББА

3.1. Постановка задачі та опис алгоритму

Нейронна мережа Хопфілда - це тип штучної нейронної мережі, винайдений Джоном Хопфілдом у 1982 році. Він зазвичай працює, спочатку вивчаючи ряд подвійних моделей, а потім повертаючи той, який найбільш схожий на заданий вхід.

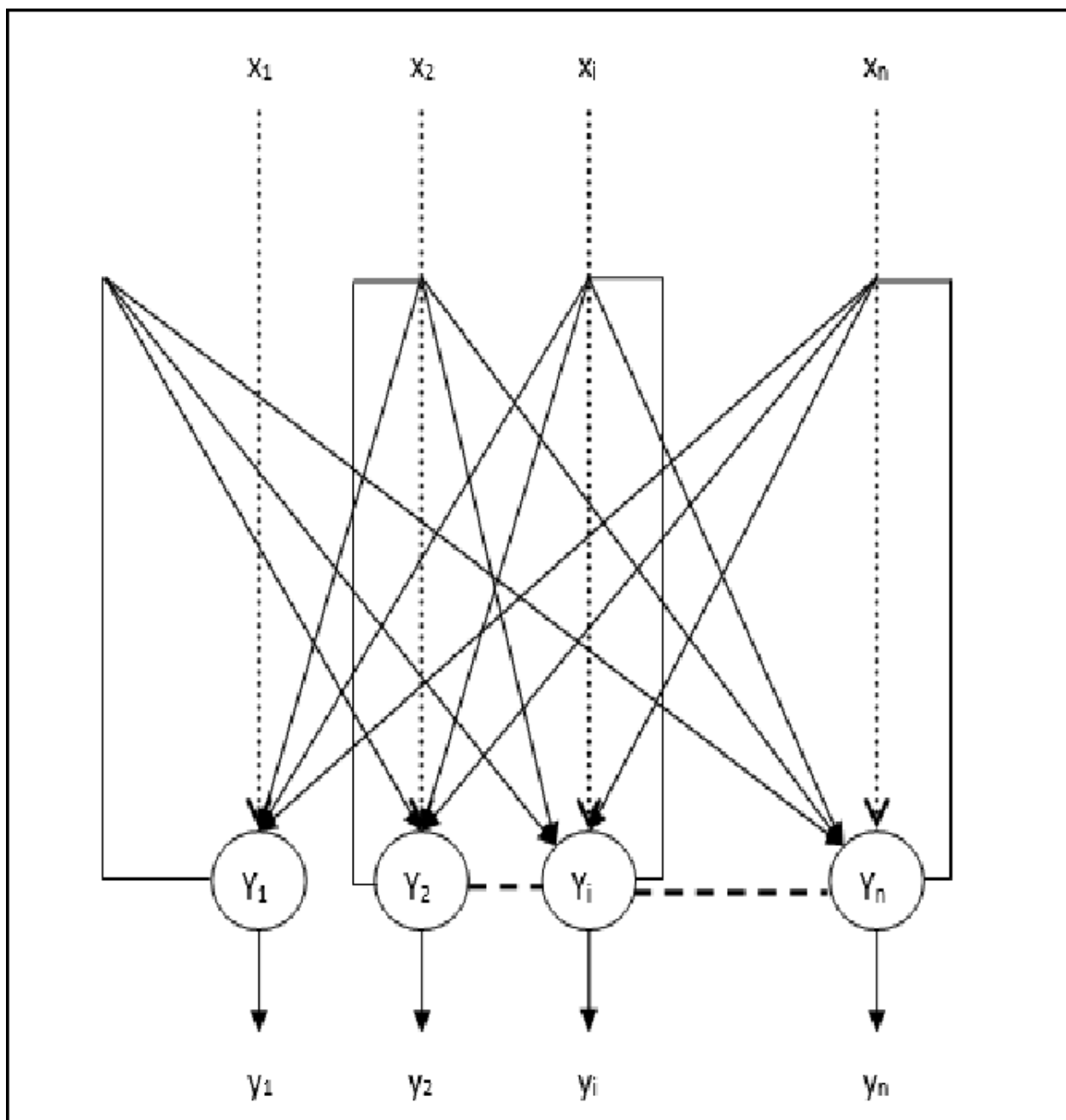


Рисунок 3.1 – Загальна схема бінарної нейронної мережі Хопфілда

Вона складається з лише одного шару вузлів або вузлів, кожен з яких пов'язаний з усіма іншими, але не з собою. Але, це мережа зворотного зв'язку, що означає, що його результати перенаправляються на його входи. Кожна одиниця також виступає як вхід і вихід мережі. Таким чином, кількість вузлів, входів, виходів мережі рівні. Крім того, кожен з нейронів має бінарне активаційне значення, зазвичай представлене як 1 або -1, що є його конкретним виходом. Стани кожного вузла взагалі сходяться, а це означає, що стан кожного вузла стає фіксованим після певної кількості оновлень.

Вузли мережі Хопфілда можуть бути оновлені або синхронно, або асинхронно. У випадку синхронно оновленої мережі годинник координує різні вузли так, щоб всі вони були оновлені одночасно. У випадку асинхронно оновленої мережі вузли оновлюються по одному, а вузол вибирається випадковим чином. Оскільки в таких біологічних системах, як мозок, для синхронізації оновлення вузлів не існує, асинхронні мережі вважаються більш реалістичними. В обох випадках існує вага з'єднання між будь-якими двома різними вузлами, оскільки всі вузли з'єднані. Вага між будь-яким даним вузлом і собою вважається 0.

Якщо вага між двома вузлами позитивна, то стани двох вузлів сходяться, інакше вони будуть розходитися. Вага може розглядатися як індикатор кінцевого стану вузла щодо кінцевого стану інших.

Існує кілька правил підготовки мережі Хопфілд, всі з яких засновані на тому ж процесі: модифікація синаптичних ваг. З урахуванням введення для вивчення, вага з'єднання між вузлами з тим же станом збільшується, тоді як вага між вузлами з протилежними станами зменшується. Правило навчання може мати дві корисних властивості: бути локальним та набувати поступовості. Воно є локальним, якщо для кожного оновлення воно використовує тільки інформацію з обох вузлів, маса яких підключена. Це відбувається поступово, якщо воно не вимагає будь-якої інформації про інші шаблони, раніше засвоєні мережею.

Теорія Хебба була введена в 1949 році Дональдом Олдіном Хеббом у своїй книзі "Організація поведінки". Ця теорія також називається постулатом Хебба або правлінням Хебба.

Реалізація правил Хебба є простою і виконується кількома кроками. Реалізація правила Хебба спочатку розглядає вхідні значення та очікувані вихідні значення, потім використовується функція активації та, нарешті, алгоритм Хебба реалізується.

Хеббейський алгоритм використовується в багатьох областях, особливо в обробці мови та зображення.

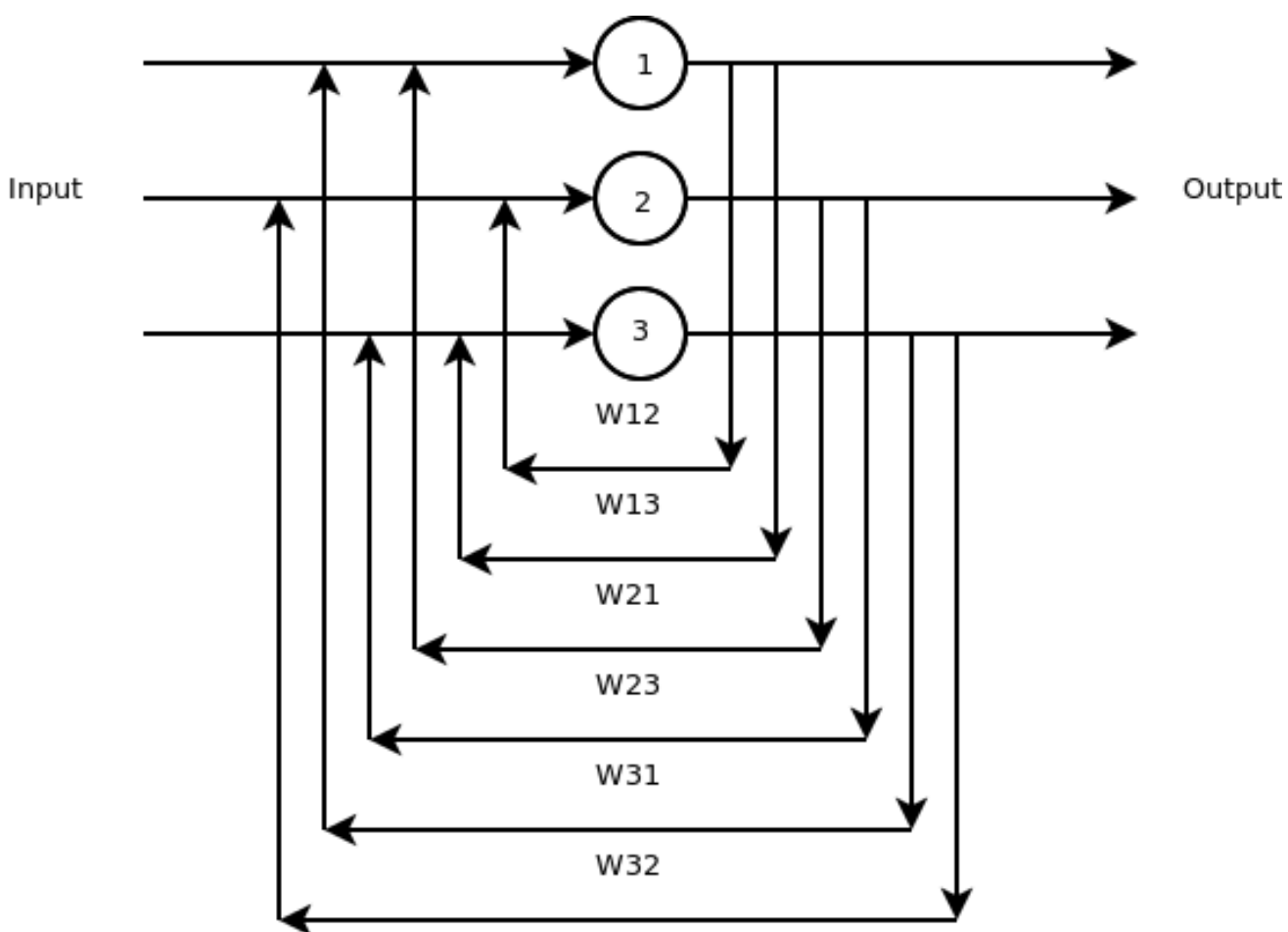


Рисунок 3.2 – Розпізнавання образу мережею Хопфілда

Питання, чи зможе така нейронна мережа повністю імітувати людський мозок в оптимальному вигляді, є досі актуальним. Нейромережі дуже корисні для програм у поєднанні з іншими компонентами, такими як генетичні алгоритми, нечітка логіка, експертні системи тощо. Всі існуючі технології нейронних мереж можуть бути вдосконалені при інтеграції з вищевказаними

компонентами для розпізнавання образів, аналізу фондових ринків та проблем прогнозування, медична діагностика пацієнтів, перетворення рукописних документів в інші відповідні формати та в інших областях, де люди мають велику кількість даних.

Метою використання асоціативної нейронної мережі є пошук відповідного вихідного вектора, який відповідає вхідному вектору, який може бути одним з збережених моделей або новим шаблоном.

Кожна асоціація являє собою пару векторів вводу-виводу, де вхідний вектор, і є цільовим вектором. Якщо обидва вектора і є однаковими, то ми називаємо це - нейронні мережі автоасоціативної пам'яті. Якщо вектори різні, то це називається нейронною мережею гетероасоціативної пам'яті.

Правило Хебба є звичайним і простим методом для визначення ваг для нейронної мережі асоціативної пам'яті. Правило Хебба може бути використане з будь-яким двійковим або біполярним патерном.

Відповідність правила Хебба в різних проблемах полягає в співвідношенні між векторами тренувального входу. У випадку, якщо вхідні вектори ортогональні (якщо їх точка продукту дорівнює 0), правильна вага буде вироблятися за правилом Хебба.

3.2. Розробка нейромережі Хебба

Залежно від того, задається чи відповідність між структурними описами апіорно або встановлюється в процесі розпізнавання, будемо розрізняти два способи застосування СІМ: жорсткий і гнучкий. Жорсткий спосіб передбачає задану апіорно фіксовану структуру безлічі фрагментів об'єкта (ідентичну еталону). При гнучкому підході структура об'єкта підлягає встановленню в процесі розпізнавання. Перевагою другого способу (модель у вигляді безлічі) є велика адаптивність і інтелектуальність, тобто можливість налаштування на довільну в певних межах форму змінюється об'єкта або формування неспотвореної структури об'єкта з метою розпізнавання.

На безлічі значень функції яскравості також можна побудувати структурний опис у вигляді [8]

$$B^k(x, y) = \begin{cases} 1, & B(x, y) = k, \\ 0, & B(x, y) \neq k \end{cases}$$

для напівтонового зображення зі значенням яскравості, або у вигляді

$$B^k(x, y) = \begin{cases} k, & B(x, y) = k, \\ 0, & B(x, y) \neq k \end{cases}$$

для кольорового зображення з індексом для RGB -кольори в графічній палітрі. Обидва ці уявлення $B = \{B^k\}$ відносяться до жорсткого способу опису, тому що вони строго прив'язані до рівня яскравості або індексу кольору.

У завданнях розпізнавання на основі описів у вигляді безлічі ХП із застосуванням СІМ можна розглянути два принципово різних класи методів: засновані на аналізі окремих ХП, а також на використанні інтегральних характеристик безлічі ХП. Ідеальним для обох варіантів є випадок, коли дескриптори ХП різних класів мають відмінні розподілу. У такій ситуації можна застосувати класифікацію на основі таких характеристик класів, як математичне очікування, дисперсія та ін.[6, 15]. Як показують матеріали попередніх розділів, для ХП візуальних об'єктів ця проста ситуація зовсім не характерна. Тут, навпаки, маємо справу з описами, досить близькими для різних класів [61].

Розділ присвячений розробці та узагальненню моделей подібності при розпізнаванні візуальних об'єктів шляхом зіставлення їх структурних описів на підставі голосування, а також синтезу ієрархічного методу розпізнавання в умовах складного фону шляхом побудови інтегрального опису із застосуванням одновимірної інтегрально-диференціальної узгодженої обробки і визначення групових ознак. Основна мета проведених досліджень - формалізація розпізнавання об'єктів на основі голосування із застосуванням апарату теорії множин. Це досягається теоретико-множинним поданням процедур голосування, розробкою нових і модифікацією класичних методів

зіставлення ХП, побудовою відносин для ХП, обґрунтуванням ймовірнісної моделі прийняття рішень на основі функції відповідності ознак. Також було здійснено аналіз шляхів практичного застосування і підтверджена ефективність запропонованих методів.

Ключові напрямки при побудові СІМ з потрібними властивостями:

- а) посилення відмінностей в просторі дескрипторів [8];
- б) використання багаторівневого аналізу на основі відносин [4] ;
- в) відбраковування «помилкових» дескрипторів в процесі обчислення структурного подібності [9] та інші.

Flow chart of Hebb training algorithm

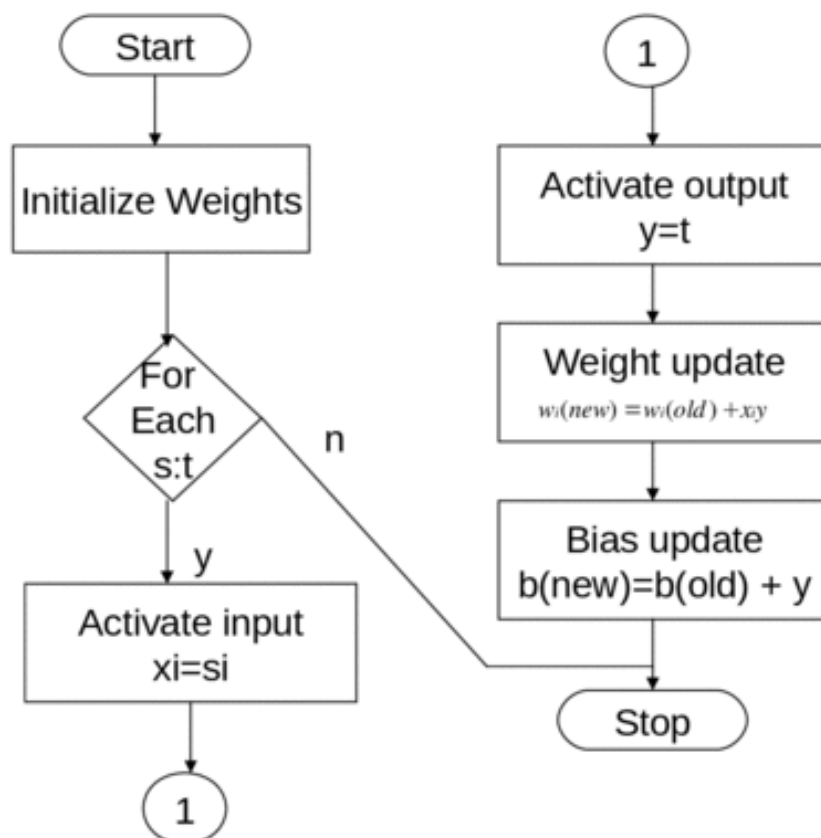


Рисунок 3.3 – Загальна схема алгоритму Хебба

У ситуаціях порівняно високого рівня сигнал-шум розпізнавання об'єктів на складному тлі можна здійснити шляхом побудови інтегральних

ознак об'єктів [6]. Розглянемо підхід, в якому аналізуються ділянки у вигляді фрагментів рядків і стовпців на предмет відповідності елементам об'єкта. Формуються локальні ознаки, за якими згодом на основі просторових відносин будуються глобальні опису. Розпізнавання і обчислення параметрів здійснюється з глобальних уявленням. На відміну від МЧК тут структура розбиття на фрагменти не фіксована, тобто ХП формуються незалежно. В результаті оцінюється структура об'єкта допускає гнучкість, що задається правилом розпізнавання.

Узагальнимо модель заходів подібності для зіставлення множин ХП при розпізнаванні шляхом голосування. В задачах розпізнавання об'єктів, описуваних наборами ХП, вихідна інформація може надаватися у вигляді кінцевого безлічі дескрипторів. По суті, такий структурний опис є не вектором, а безліччю, тому що порядок проходження елементів залежить від процесу обробки. Тому оцінка подібності (близькості) повинна розглядатися як міра на множинах, а завдання віднесення об'єкта до класу вирішується шляхом оптимізації близькості множин ХП об'єкта й еталона [9].

Для оцінки подібності в умовах просторових перешкод застосовні не всі класичні заходи і метрики для множин. Наприклад, метрики Хаусдорфа і ближнього (дальнього) сусіда фактично формуються за величиною одного з відстаней між елементами і з цієї причини не можуть вважатися стійкими до спотворень. З іншого боку, інтегрально враховує схожість всіх без винятку елементів величина «середньої зв'язку» [3] має гарну помехозащищенностью щодо адитивних перешкод, однак, до спотворень ХП вона також нестійка. Тому виникає необхідність в модифікації наявних заходів або в розробці нових заходів близькості. Найбільш близькі до задоволення необхідних властивостей заходи, пов'язані з множинними операціями перетину і симетричної різниці, які можуть бути обчислені шляхом підрахунку відповідностей схожих елементів. Підрахунок еквівалентних ХП аппроксимує оцінку подібності об'єктів.

Вдало побудована міра повинна не тільки знизити вплив похибок через незначних відхилень при дії адитивних перешкод, але і шляхом відкидання помилкових відповістей дескрипторів забезпечити надійне розпізнавання об'єктів при істотних змінах через дії просторових перешкод [49, 86, 87].

При роботі з множинами оперують поняттям «міра безлічі» [6, 17]. Суть заходи для кінцевих множин полягає в оцінці потужності (кількості елементів). Обчислення заходи перетину множин зводиться до підрахунку числа «загальних» елементів. З цієї точки зору такі загальноприйняті заходи як перетин і симетрична різниця мають близьку трактування.

При вирішенні проблем аналізу двовимірних сигналів і оцінки біометричної інформації розрізняють три основних типи заходів подібності [7-9]:

- асоціативні, що виражають відносини числа співпадаючих ознак до загального їх числа (коефіцієнти зв'язку, парні функції);
- кореляційні («косинусні» заходи);
- відстані в метричному просторі.

Вибір заходів подібності залежить в першу чергу від мети застосування. Всі заходи, по суті, пов'язані між собою, тому що в кінцевому підсумку оцінюють ймовірність події ідентичності порівнюваних об'єктів в сенсі володіння загальними ознаками або атрибутами [8, 9]. Заходи подібності можна розділити на метрики і міри схожості. Міра подібності повинна бути невід'ємною, симетричною, максимальною при порівнянні об'єкта з самим собою [9].

Штучні нейронні мережі використовують різні шари математичної обробки, щоб мати сенс інформації, яку вона годує. Як правило, штучна нейронна мережа має десь від десятків до мільйонів штучних нейронів, так званих одиниць, розташованих у серії шарів. Вхідний шар отримує різні види інформації від зовнішнього світу. Це дані, якими мережа націлена на обробку або вивчення. З блоку введення дані проходять через один або декілька

прихованих блоків. Робота прихованої одиниці полягає в тому, щоб перетворити вхід на те, що може використовувати пристрій виводу.

Більшість нейронних мереж повністю з'єднані від одного шару до іншого. Ці з'єднання зважуються; чим вище число, тим більший вплив має одна одиниця на інший, подібний до людського мозку. Оскільки дані проходять через кожен блок, мережа більше дізнається про дані. На іншій стороні мережі є одиниці виводу, і саме там мережа реагує на дані, які їй було надано та оброблено.

Когнітивні нейробіологи вивчили величезну кількість людського мозку, оскільки комп'ютерні вчені спочатку спробували оригінальну штучну нейронну мережу. Одне з речей, які вони дізналися, полягає в тому, що різні частини мозку відповідають за обробку різних аспектів інформації, і ці частини розташовані ієрархічно. Отже, вхід входить у головний мозок, і кожен рівень нейронів забезпечує розуміння, а потім інформація передається на наступний, більш високий рівень. Саме такий механізм, який нейронні мережі намагаються відтворити.

Для того, щоб нейронна мережа навчалася, вони повинні мати величезну кількість інформації, яку вони вираховують, називається навчальним набором. Коли ви намагаєтеся навчити ANN, як відрізнити кішку від собаки, набір тренувань забезпечить тисячі зображень, позначених як собака, щоб мережа почала навчатися. Після того, як він отримав навчання з великою кількістю даних, він буде намагатися класифікувати майбутні дані на основі того, що він думає, що він бачить (або слухання, залежно від набору даних) у різних одиницях. Протягом періоду тренувань продуктивність машини порівнюється з описом того, що слід спостерігати, наданим людьми. Якщо вони однакові, машина перевіряється. Якщо це неправильно, вона використовує зворотне розповсюдження, щоб налаштувати своє навчання, повертаючись через шари, для налаштування математичного рівняння. Відоме як глибоке вивчення, це те, що робить мережу інтелектуальною.

Існує кілька способів розгортання штучних нейронних мереж, включаючи класифікацію інформації, прогнозування результатів та дані кластеру. Оскільки мережі обробляють і вивчають дані, вони можуть класифікувати певний набір даних у попередньо визначений клас, його можна навчити прогнозувати результати, які очікуються від даного вводу, і можуть ідентифікувати особливу властивість даних, щоб потім класифікувати дані.

Комп'ютери мають можливість розуміти навколишній світ у дуже людському вигляді завдяки силі штучних нейронних мереж.

Кожен процесор подібної мережі має справу тільки з сигналами, які він періодично отримує, і сигналами, які він періодично посилає іншим процесорам. І, тим не менше, будучи з'єднаними в досить велику мережу з керованою взаємодією, такі окремо прості процесори разом здатні виконувати досить складні завдання.

- З точки зору машинного навчання, нейронна мережа являє собою окремий випадок методів розпізнавання образів, дискримінантного аналізу, методів кластеризації та інше.

- З математичної точки зору, навчання нейронних мереж – це багатопараметрична завдання нелінійної оптимізації.

- З точки зору кібернетики, нейронна мережа використовується в задачах адаптивного управління і як алгоритми для робототехніки.

- З точки зору розвитку обчислювальної техніки та програмування, нейронна мережа – спосіб вирішення проблеми ефективного паралелізму.

А з точки зору штучного інтелекту, ІНС є основою філософської течії коннективізма і основним напрямком в структурному підході з вивчення можливості побудови (моделювання) природного інтелекту за допомогою комп'ютерних алгоритмів.

Нейронні мережі не програмуються в звичному сенсі цього слова, вони навчаються. Можливість навчання – одне з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в

знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних і/або «зашумлених», частково спотворених даних.

Апріорну щільність ймовірності можна оцінити різними способами. У параметричних методах передбачається, що щільність ймовірності (PDF) є функцією певного виду з невідомими параметрами. Наприклад, можна спробувати наблизити PDF за допомогою гауссових функцій. Для того щоб зробити класифікацію, потрібно попередньо отримати оціночні значення для вектора середнього і матриці коваріацій по кожному з класів даних і потім використовувати їх у вирішальному правилі.

Вся описана процедура називається квадратичним дискримінантний аналізом (QDA). У припущенні, що матриці коваріацій у всіх класів QDA зводиться до лінійного дискримінантного аналізу

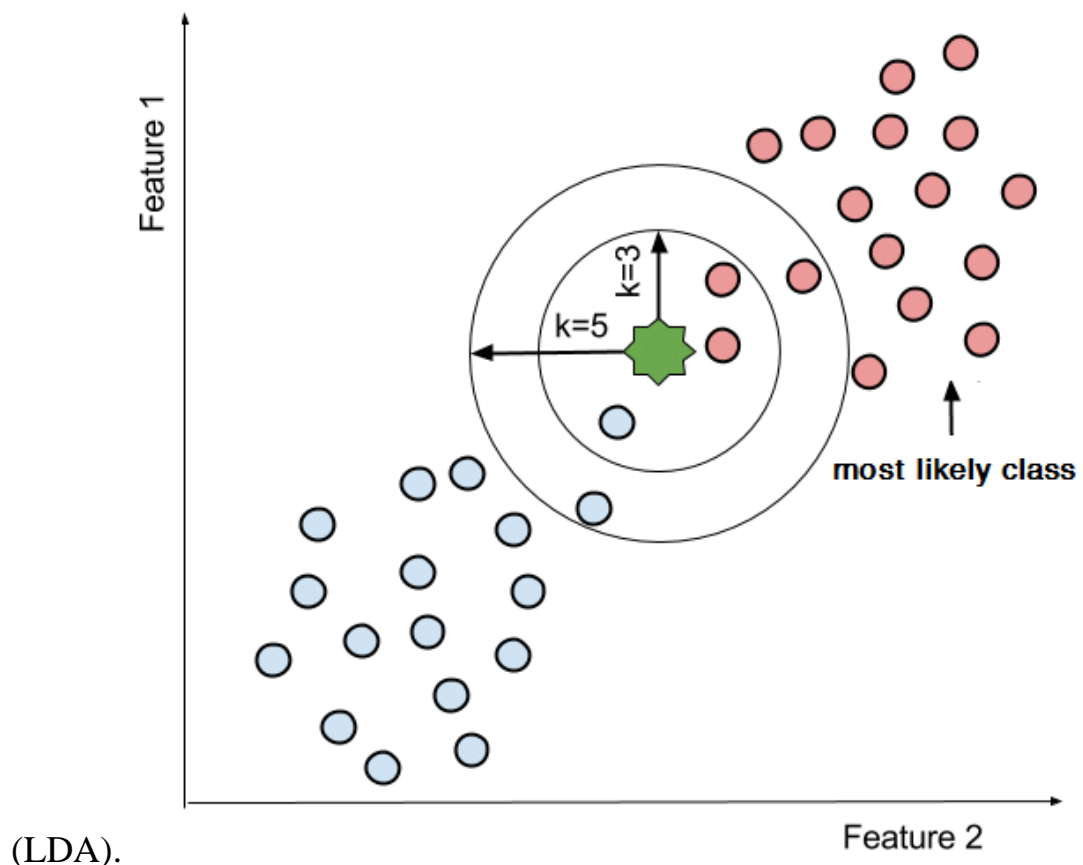


Рисунок 3.4 – Загальна схема KNN моделі

У іншого типу - непараметричних - ніяких попередніх припущень про щільність ймовірності не потрібно. У методі «к найближчих сусідів» (kNN) обчислюється відстань між знову надійшли зразком і векторами навчальної множини, після чого зразок відноситься до того класу, до якого належить більшість з k його найближчих сусідів. В результаті цього кордону, що розділяють класи, виходять кусочно-лінійними. У різних модифікаціях цього методу використовуються різні заходи відстані і спеціальні прийоми знаходження сусідів. Іноді замість самого безлічі зразків береться сукупність Центроїд, відповідних кластерів у методі адаптивного векторного квантування (L VQ).

В інших методах класифікатор розбиває дані на групи за схемою дерева. На кожному кроці підгрупа розбивається надвоє, і в результаті виходить ієрархічна структура бінарного дерева. Цей метод хороший тим, що він породжує метод класифікації, заснований на логічних вирішальних правилах. Ідеї деревовидних класифікаторів застосовуються в методах побудови самонарашуваних нейронних класифікаторів.

Нехай є безліч M зображень, для яких відома коректна класифікація на два класи $X_1 = \{X_{11}, X_{12}, \dots, X_{1q}\}$, $X_2 = \{X_{21}, X_{22}, \dots, X_{2p}\}$, , , І нехай першого класу X_1 відповідає вихідний сигнал $y = 1$, а класу X_2 - сигнал $y = -1$. Якщо, наприклад, пред'явлено деяке зображення і його зважена сума вхідних сигналів перевищує нульове значення:

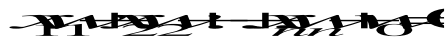
$$S = \sum_{i=1}^n x_i w_i > 0$$

то вихідний сигнал $y = 1$ і, отже, вхідне зображення X_α належить класу X_1 . Якщо $S < 0$, то $y = -1$ і пред'явлене зображення належить другому класу.

Можливе використання окремого нейрона і для виділення з безлічі класів $M = \{X_1 = \{X_{11}, \dots, X_{1k}\}, \dots, X_i = \{X_{i1}, \dots, X_{iq}\}, \dots, X_p = \{X_{p1}, \dots, X_{pt}\}\}$ зображень єдиного класу X_i . В цьому випадку вважають, що один з двох можливих вихідних сигналів нейрона (наприклад, 1) відповідає класу X

і а другий - всім іншим класам. Тому, якщо вхідне зображення $X \in X_i$ призводить до появи сигналу $y = 1$, то $X \in X_i$, якщо $y = -1$ (або $y = 0$, якщо використовується бінарне кодування), то це означає, що пред'явлене зображення не належить його виділяє класу.

Система розпізнавання на основі єдиного нейрона ділить весь простір можливих рішень на дві області за допомогою гіперплощини.



Для двовимірних вхідних векторів кордоном між двома класами зображень є пряма лінія: вхідні вектори, розташовані вище цієї прямої, належать до одного класу, а нижче - до іншого.

Для адаптації, настройки або навчання ваг зв'язків нейрона може використовуватися кілька методів. Розглянемо один з них, який отримав назву "правило Хебба". Хебб, досліджуючи механізми функціонування центральної нервової системи, припустив, що навчання відбувається шляхом посилення зв'язків між нейронами, активність яких збігається за часом. Хоча в біологічних системах це припущення виконується далеко не завжди і не вичерпує всіх видів навчання, однак при навчанні одношарових нейронмереж з біполярними сигналами воно досить ефективно.

Відповідно до правила Хебба, якщо пред'явленим біполярному зображенню $X = (x_1, \dots, x_n)$ відповідає неправильний вихідний сигнал, то ваги зв'язків нейрона адаптуються по формулі

$$w_i(t+1) = w_i(t) + x_i y$$

де $w_i(t)$, $w_i(t+1)$ відповідно вага i -й зв'язку нейрона до і після адаптації; x_i - Компоненти вхідного зображення; $y = 1$ - сигнал зміщення; $y = -1$ - вихідний сигнал нейрона.

У більш повної і суворій формі алгоритм настройки ваг зв'язків нейрона з використанням правила Хебба виглядає наступним чином:

Крок 1. Здається безліч $M = \{(X_1, t_1), \dots, (X_T, t_m)\}$, що складається з пар (вхідне зображення, Необхідний вихідний сигнал нейрона t_k). Ініціюються ваги зв'язків нейрона:

$$v_i = 0 \quad i = \overline{0, n}$$

Крок 2. Для кожної пари (X_k, t_k) , поки не дотримуються умови зупинки, виконуються кроки 3-5.

Крок 3. Ініціюється безліч входів нейрона:

$$x_i = 1 \quad x_i = \overline{0, n}$$

Крок 4. Ініціюється вихідний сигнал нейрона: $y = t_k$.

Крок 5. Коректуються ваги зв'язків нейрона за правилом

$$w_{ij} = w_{ij} + \eta (t_k - y) x_i$$

Крок 6. Перевірка умов зупинки.

Якщо вектор (y_1, \dots, y_t) розрахованих вихідних сигналів дорівнює вектору (t_1, \dots, t_m) заданих сигналів нейрона, тобто кожному вхідному зображенню відповідає заданий вихідний сигнал, то обчислення припиняються (перехід до кроку 7), якщо ж $(y_1, \dots, y_t) \neq (t_1, \dots, t_m)$, то перехід до кроку 2 алгоритму.

Крок 7. Зупинка.

Використання групи з t біполярних або бінарних нейронів A_1, \dots, A_t (рис. 4) дозволяє істотно розширити можливості нейронної мережі і розпізнавати до 2^t різних зображень. Правда, застосування цієї мережі для розпізнавання 2^t (або близьких до 2^t чисел) різних зображень може призводити до нерозв'язних проблем адаптації ваг зв'язків нейромережі.

Вхідний рівень нейронів виконує розподільчі функції (input layer). Вихідний рівень здійснює обробку інформації від попередніх рівнів та видає результати. Всі інші рівні між вхідним і вихідним називаються проміжними або схованими рівнями. Ці рівні також здійснюють обробку інформації.

Кожен нейрон схованого рівня з'єднаний зі всіма виходами нейронів попереднього рівня та входами нейронів наступного рівня. Таким чином БНМ мають однорідну та регулярну структуру. Вихідне значення БНМ в загальному випадку обчислюється за формулою:

$$Y = F(F(F(x * W(1)) * W(2))) * W(3)$$

x – вектор вхідних значень,

F – оператор лінійного перетворення.

Кількість рівнів НМ визначають яким чином вхідний простір може бути розбитий на вихідні підпростори меншої розмірності.

Наприклад.

В двох рівневій НМ з одним рівнем лінійних нейронних елементів розділяє в простір вхідних образів за допомогою гіперплощини.

Трьох рівнева НМ: в якості нейронів останніх рівнів використовуються нейрони з нелінійною функцією активації, що дозволяє формувати будь – які випуклі поверхні в просторі рішень.

Розглянемо НМ, яка складається з 4-х слоїв (рівнів).

Вихідне значення j – ого нейрона останнього рівня рівна $F(S_j)$, де

$$S_j = \sum_i y_i * \omega_{ij} - T_j$$

S_j – зважена сума j – ого нейрона вихідного рівня,

y_i – вихідне значення i – ого нейрона передостаннього рівня,

ω_{ij} та T_j – вагові коефіцієнти та поріг j – ого нейрону останнього рівня.

Аналогічним чином вихідні значення i – ого нейрону передостаннього рівня визначається як:

$$y_i = F(S_i), \text{ де } S_i = \sum_k \omega_{ki} * y_k - T_i$$

Тоді для k – ого рівня:

$$y_k = f(S_k)$$

$$S_k = \sum_l y_l * \omega_{lk} - T_k$$

Алгоритм зворотної розповсюдженої помилки мінімізує середньоквадратичну помилку НМ. Для налаштування (зміни) вагових коефіцієнтів та порогів НМ використовують метод градієнтного спуску:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \frac{\partial E}{\partial \omega_{ij}(t)}$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}$$

E – середньоквадратична помилка НМ для 1 – ого навчального вектора.

Вона визначається наступним чином:

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2$$

T_j – еталонне вихідне значення j – ого нейрону.

Помилка j – ого нейрону вихідного рівняння:

$$\varphi_i = y_i - d_i$$

Помилка i – ого нейрону для схованого рівня i визначається через помилки нейронів наступного рівня j :

$$\varphi_i = \sum_j^m y_j F'(S_j) \omega_{ij}$$

Похідні середньоквадратичної помилки по вагових коефіцієнтах i порогах НМ для будь – яких двох рівнів ij визначаються наступним чином:

$$\frac{\partial E}{\partial \omega_{ij}} = \varphi_i F'(S_j) \omega_{ij}$$

$$\frac{\partial E}{\partial T_j} = \varphi_i F'(S_j)$$

S_j – зважена сума входів,

F – функція активації нейронів,

φ_j – помилка нейронів j – ого рівня.

Для мінімізації середньоквадратичної помилки НМ вагові коефіцієнти та пороги нейронів повинні змінюватись наступним чином:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha * \varphi_j * F'(S_j) * y_i$$

$$T_j(t+1) = T_j(t) + \alpha * \varphi_j * F'(S_j)$$

α - крок навчання.

Рішення завдання класифікації є одним з найважливіших застосувань нейронних мереж. Завдання класифікації представляє собою завдання віднесення зразка до одного з декількох попарно непересічних множин. Прикладом таких завдань може бути, наприклад, завдання визначення кредитоспроможності клієнта банку, медичні завдання, в яких необхідно

визначити, наприклад, результат захворювання, рішення задач управління портфелем цінних паперів (продати, купити або «притримати» акції в залежності від ситуації на ринку), задача визначення життєздатних несхильних до банкрутства фірм.

Example

$$\begin{array}{ccc}
 \text{Banana} & \text{Apple} & \text{Normalized Prototype Patterns} \\
 \mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} & \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} & \left\{ \mathbf{p}_1 = \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, \mathbf{t}_1 = [-1] \right\} \quad \left\{ \mathbf{p}_2 = \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, \mathbf{t}_2 = [1] \right\}
 \end{array}$$

Weight Matrix (Hebb Rule):

$$\mathbf{W} = \mathbf{T}\mathbf{P}^T = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} -0.5774 & 0.5774 & -0.5774 \\ 0.5774 & 0.5774 & -0.5774 \end{bmatrix} = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix}$$

Tests:

$$\text{Banana} \quad \mathbf{W}\mathbf{p}_1 = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} -0.6668 \end{bmatrix}$$

$$\text{Apple} \quad \mathbf{W}\mathbf{p}_2 = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix} \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} 0.6668 \end{bmatrix}$$

Рисунок 3.5 – Приклад застосування правила Хебба

При вирішенні задач класифікації необхідно віднести наявні статичні зразки (характеристики ситуації на ринку, дані медогляду, інформація про клієнта) до певних класів. Можливі кілька способів подання даних. Найбільш поширеним є спосіб, при якому зразок видається вектором. Компоненти цього вектора є різні характеристики зразка, які впливають на прийняття рішення про те, до якого класу можна віднести даний зразок. Наприклад, для медичних завдань в якості компонентів цього вектора можуть бути дані з медичної карти хворого. Таким чином, на підставі деякої інформації про приклад, необхідно визначити, до якого класу його можна віднести. Класифікатор таким чином відносить об'єкт до одного з класів відповідно до

визначеного розбиттям N-мірного простору, яке називається простором входів, і розмірність цього простору є кількістю компонент вектора.

Перш за все, потрібно визначити рівень складності системи. В реальних задачах часто виникає ситуація, коли кількість зразків обмежено, що ускладнює визначення складності завдання. Можливо виділити три основні рівні складності.

Перший (найпростіший) - коли класи можна розділити прямими лініями (або гіперплоскостями, якщо простір входів має розмірність більше двох) - так звана лінійна роздільність.

У другому випадку класи неможливо розділити лініями (площинами), але їх, можливо, відокремити за допомогою більш складного поділу - нелінійна роздільність.

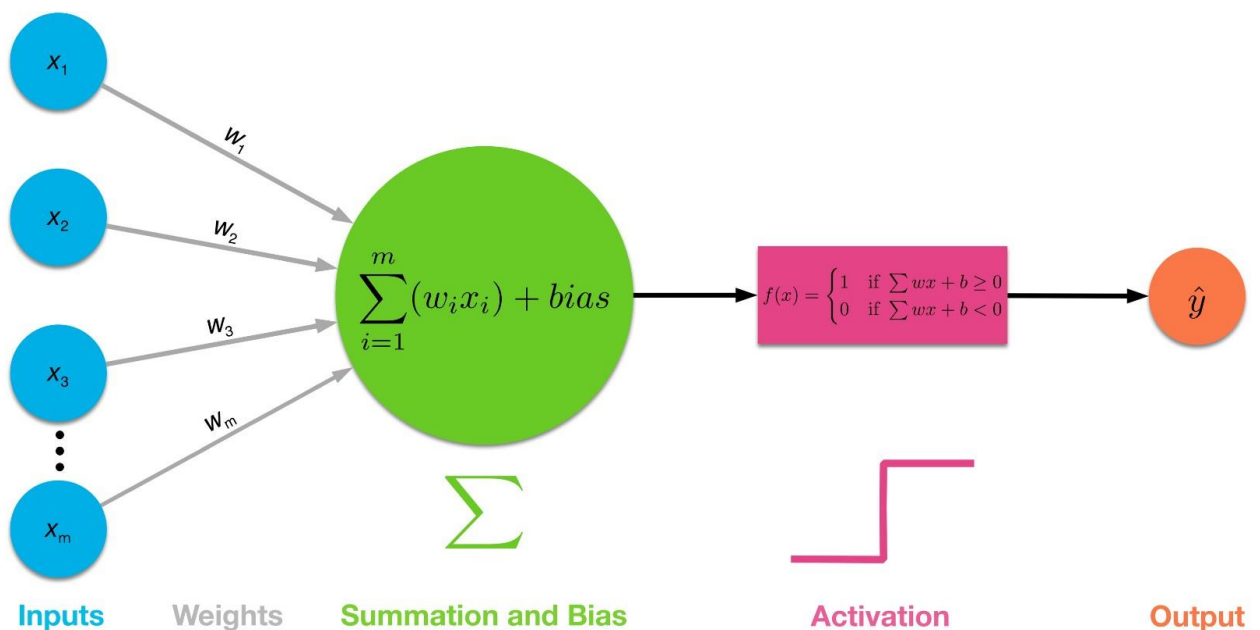


Рисунок 3.6 – Загальна модель персептрона

Розглядаються методи навчання штучних дійсних нейромереж, побудованих із НЕ із згладженими двопороговими функціями активації та комплексних нейромереж. Розроблено нейромережеву модель для задачі прогнозування обсягу замовлень та створено на її основі інформаційну прикладну програмну систему. На результатах комп'ютерних експериментів продемонстровано переваги застосування ШНМ із згладженими двопороговими функціями активації. Досліджено властивості комплексних

нейронних елементів з узагальненими функціями активації. Наведено модифікацію методу зворотного поширення похибки навчання комплексних штучних нейронних мереж.

З використанням нейромережевого моделювання було розроблено прикладну програмну систему прогнозування обсягу замовлень на деякі види продукції УАП ТОВ «Фішер-Мукачево». На основі аналізу сезонного характеру попиту на бігові лижі та маркетингових досліджень було прийнято рішення включити у нейромережеву модель наступні впливові фактори:

- 1) середньомісячна температура повітря у вибраних країнах протягом трьох місяців зимового сезону;
- 2) індекс споживчої довіри у вибраних країнах протягом місяців зимового сезону;
- 3) ціна одиниці продукції кожного найменування.

При формуванні навчальної вибірки моделі використовуються дані про значення впливових факторів 1-2 для наступних країн: Німеччина, Росія (Центральний федеральний округ), Норвегія, Швеція, Фінляндія.

Для моделювання використовується багатошарова ШНМ прямого поширення, структура якої наведена на рис. 1. Мережа побудована з штучних нейронів з лінійним входним оператором. Нейрони першого шару мають 31 вхід. Вихідний шар містить 1 нейрон. Кількість проміжних прихованих шарів та кількість нейронів у кожному з цих шарів може змінюватися. При моделюванні використовувалися як НЕ з сигмоподібними функціями активації, так і НЕ зі згладженими двопороговими функціями активації

$$y = 1 - 2e^{-x^2};$$

$$y = \frac{2}{1 + e^{-10(x-1)}} - \frac{2}{1 + e^{-10(x+1)}} + 1.$$

Навчальна вибірка $\{(\mathbf{x}^t, d^t)\}$, $t=1, \dots, 12$ сформована на основі значень впливових факторів \mathbf{x}^t та скалярних величин d^t – обсягів замовлень у відповідному році. Вихід мережі на вході \mathbf{x}^t позначається $f^t = f(\mathbf{x}^t)$.

Оскільки для вибраних функцій активації нейронів вихідне значення мережі потрапляє у відрізок $[-1; 1]$, то перед початком процесу навчання потрібно проводити масштабування елементів навчальної вибірки, альтернативою до якого є використання ШНМ, нейрони останнього шару якої мають лінійні функції активації. Функція помилки мережі має вигляд:

$$E = \frac{1}{2} \sum_t (f^t - d^t)^2$$

У процесі навчання на кожній ітерації ваговий вектор корегується у напрямку антиградієнта E : $\mathbf{w}^{r+1} = \mathbf{w}^r + \Delta \mathbf{w}^r$, де r – номер ітерації і $\Delta \mathbf{w}^r = -\eta_r \text{grad } E(\mathbf{w}^r)$.

Формула (10) використовується для обчислення похибки у випадку offline-навчання. У режимі online-навчання при обчисленні функції похибки сумування не проводиться і в якості похибки мережі використовується величина $E = 0,5 \cdot (f^t - d^t)^2$.

Розглядаються штучні нейронні мережі з комплексними вагами та їх навчання. Неперервний комплексний нейронний елемент (нейрон з неперервною комплексною функцією активації) (НКНЕ) – функціональний елемент із n входами та одним виходом *out*, який обчислюється наступним чином:

$$out = f \left(\sum_{j=1}^n w_j z_j + w_0 \right),$$

де комплексні числа z_1, \dots, z_n – вхідні сигнали, w_1, \dots, w_n – вагові комплексні коефіцієнти, w_0 – зміщення та $f: \mathbb{C} \rightarrow \mathbb{C}$ – функція активації, неперервна разом із своїми частинними похідними за дійсною і уявною частинам аргументу. В роботі розглядаються багатошарові комплексні штучні нейронні мережі (КШНМ) прямого поширення на основі, побудовані на основі НКНЕ.

Надалі будемо використовувати наступні позначення: індекс j позначає номер вхідного нейрона, k – номер вихідного нейрона, l – номер шару,

$z_{jkl} = x_{jkl} + i y_{jkl}$ – значення j -го вхідного сигналу k -го нейрона в шарі l ,
 $w_{jkl} = u_{jkl} + i v_{jkl}$ – значення j -го вагового коефіцієнта k -го нейрона в шарі l .

Багатошарова комплексна штучна нейронна мережа обчислює вихідний вектор $F(\mathbf{z})$ на основі вхідного вектора \mathbf{z} . У процесі навчання змінюємо вагові коефіцієнти w_{jkl} таким чином, щоб комплексна штучна нейронна мережа ставила у відповідність вхідним векторам із множини $\{\mathbf{z}^1, \dots, \mathbf{z}^m\}$ відповідні їм вихідні вектори з множини $\{\mathbf{d}^1, \dots, \mathbf{d}^m\}$ (або близькі до них). Сукупність пар $\{(\mathbf{z}^1, \mathbf{d}^1), \dots, (\mathbf{z}^m, \mathbf{d}^m)\}$ утворює навчальну вибірку.

Нехай f_k^t – значення вихідного сигналу k -го нейрона в останньому (вихідному) шарі l мережі на вході \mathbf{z}^t . Похибка мережі E обчислюється за формулою:

$$E = \frac{1}{2} \sum_k \sum_t |f_k^t - d_k^t|^2.$$

З огляду на складність завдань структурного розпізнавання, пов'язану з близькістю дескрипторів для різних об'єктів, основна увага при побудові заходів зосередимо на те, щоб на першому етапі шляхом логічної обробки оцінити «збіг» окремих ХП, а потім підрахунком прецедентів (голосів) прийняти остаточне рішення. При цьому на рівні ХП перевагу віддамо використання метрик. Приклади метрик для ХП наведені в розділі 2.

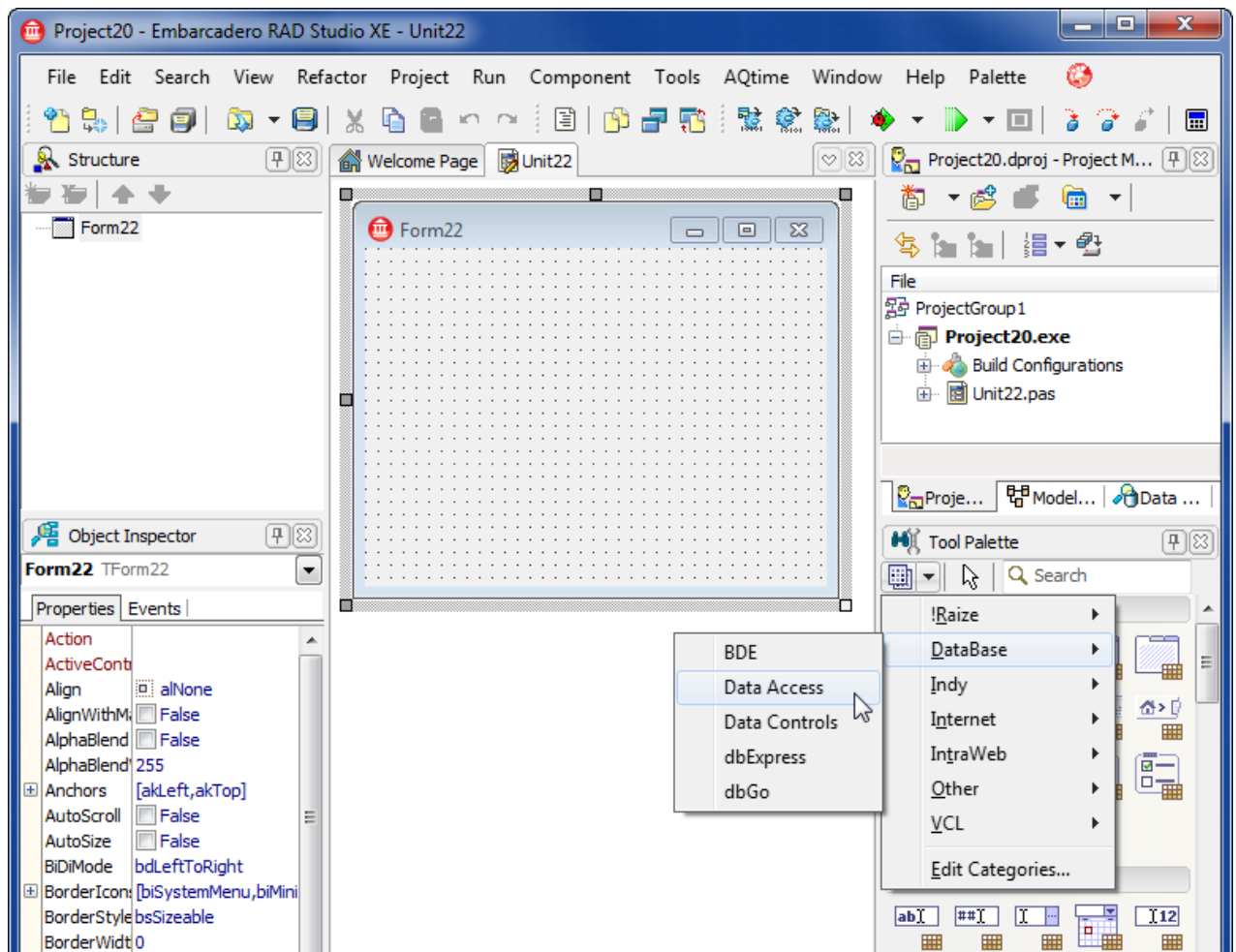


Рисунок 3.6 – IDE Delphi

Своє ім'я нова система (Delphi) отримала в честь давньогрецького міста Дельфи. Назва була вибрана неспроста: місто Дельфи пов'язаний з ім'ям бога мудрості і покровителем мистецтв Аполлона. Згідно з легендою головне святилище Аполлона знаходилося саме в цьому місті. Його жриці-сивілі звіщали пророцтва бажаючим дізнатися свою долю. «Ім'я Delphi було запропоновано Денні Торпом під час мозкового штурму. Ми хотіли, щоб в імені системи відбилися унікальні здібності продукту до роботи з базами даних, і Delphi якнайкраще перегукується з таким заслуженим ім'ям в цій області, як Oracle, принаймні для тих, кому поєднання «Дельфійський Оракул» про щось говорить », розповідав керівник дослідницької групи з розробки системи Delphi Чак Язджевські.

Основні версії. Версія 1 була призначена для розробки під 16-розрядну платформу Win16. Починаючи з другої версії, була реалізована можливість компіляції програми під 32-розрядну платформу Win32. У версії 3 з'явилася

підтримка багатоланкової технології (multi-tiered) доступу до даних, що дозволило створювати масштабовані додатки (відносно слабо залежать від сервера БД) за рахунок перенесення методів обробки інформації (бізнес-правил) на середню ланку. Разом з 6-й версією Delphi вийшла сумісна з ним по мові і бібліотекам середу Kylix, призначена для компіляції програм під операційну систему Linux. Версія 8 здатна генерувати байт-код виключно для платформи .NET. Це перша середа, орієнтована на розробку багатомовних додатків (лише для платформи .NET). Подальші версії (що позначаються роками виходу, а не порядковими номерами, як це було раніше) дозволили створювати як додатки Win32, так і .NET-додатки.

У Delphi 2006, з'явилася можливість писати програми для .NET використовуючи стандартну бібліотеку класів .NET, VCL для .NET. Середовище також дозволила писати .NET-додатки на C # і Win32-додатки на C ++. У Delphi 2006 також була реалізована технологія MDA (Model Driven Architecture) за допомогою ECO (Enterprise Core Objects) версії 3.0.

3.3. Застосування розробленої нейромережі Хебба

Застосуємо звичайний метод Хебба до розпізнавання схожих образів.

За приклад візьмемо розпізнавання схожих цифр, наприклад «3» і «8».

Запускаємо програму простого методу Хебба:

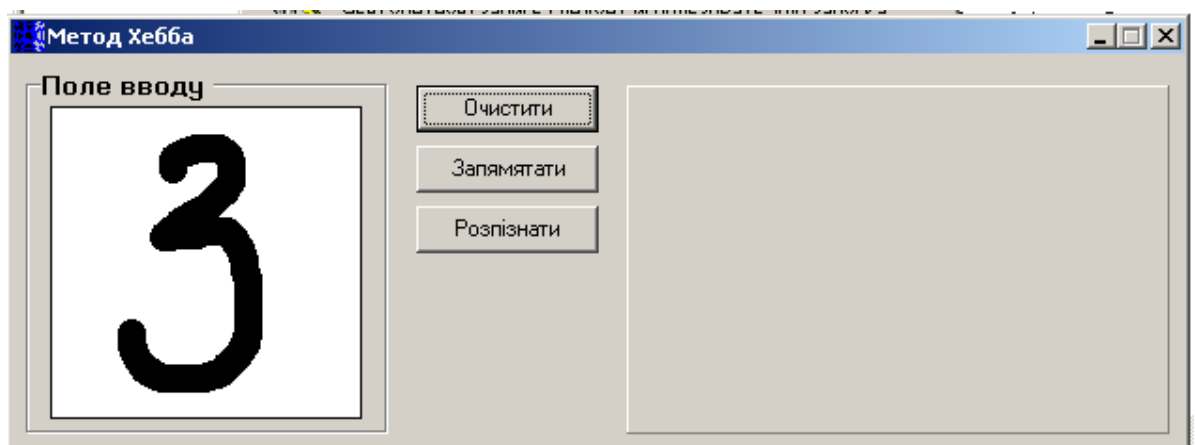


Рисунок 3.7 - Ввід цифри «3»

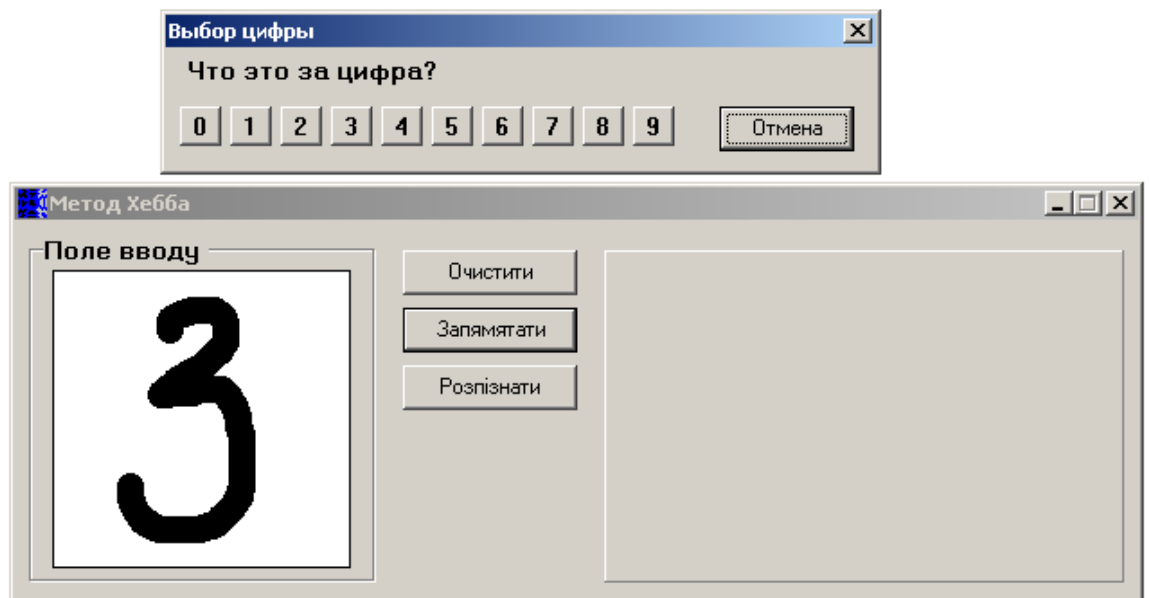


Рисунок 3.8 - Навчання цифри «3»

Аналогічні дії проводимо з цифрою «8».

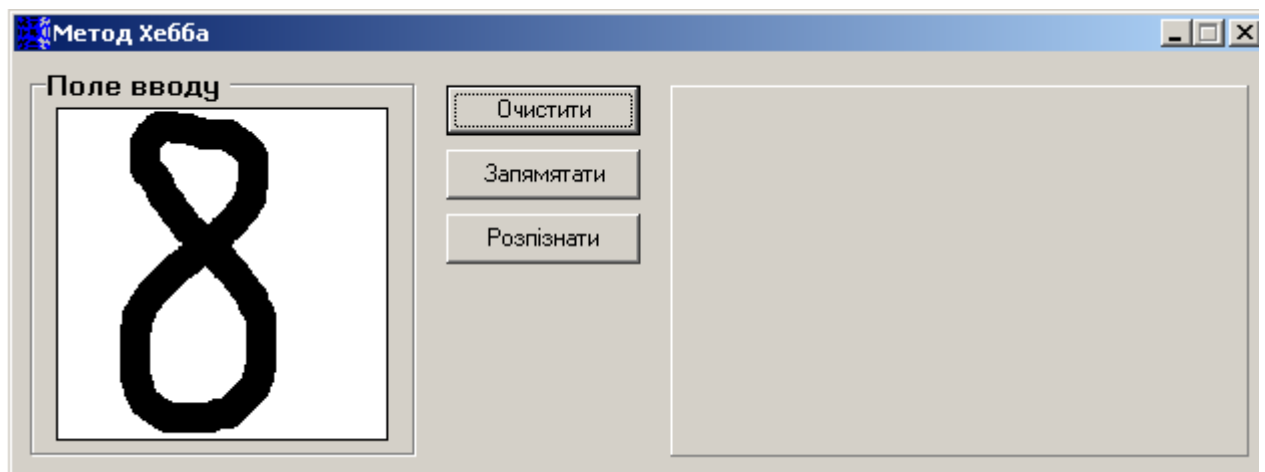


Рисунок 3.9 - Ввід цифри «8»

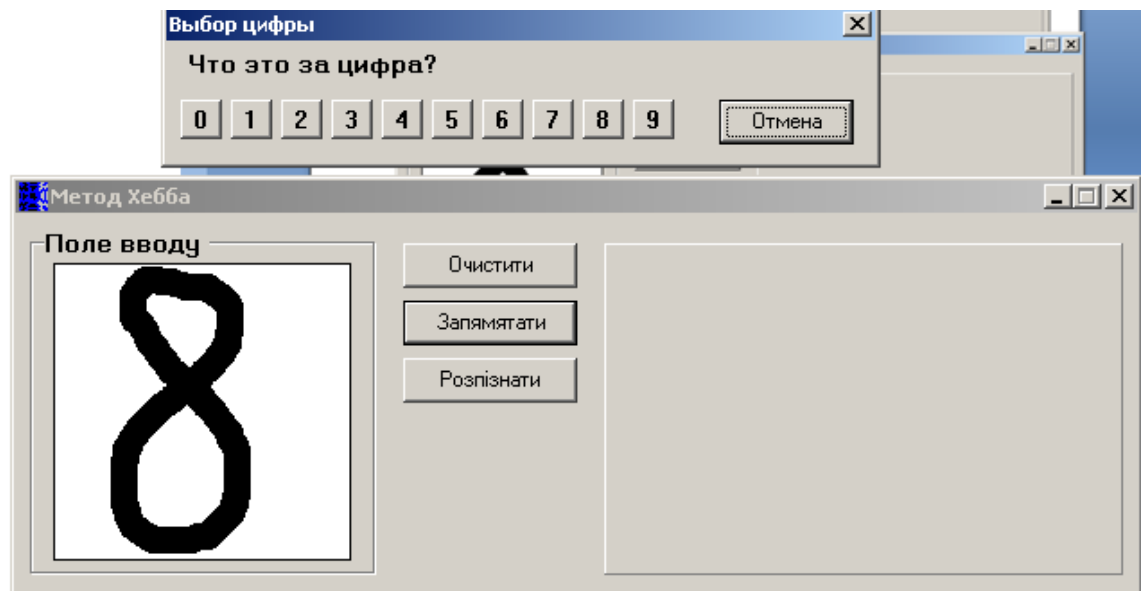


Рисунок 3.10 - Навчання цифри «8»

Далі, використовуючи простий метод Хебба розпізнаємо схожі зображення. Почнемо з цифри «3».

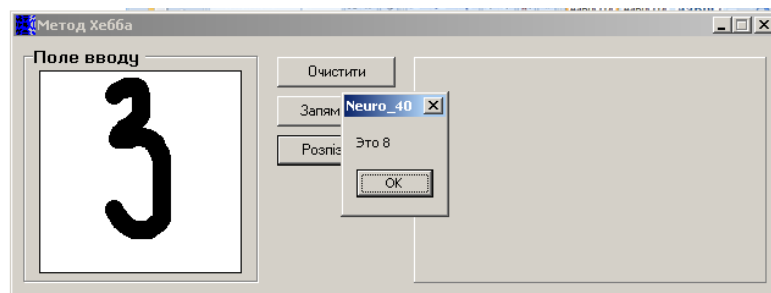


Рисунок 3.11 - Розпізнавання цифри «3»

Висновок програми не вірний.

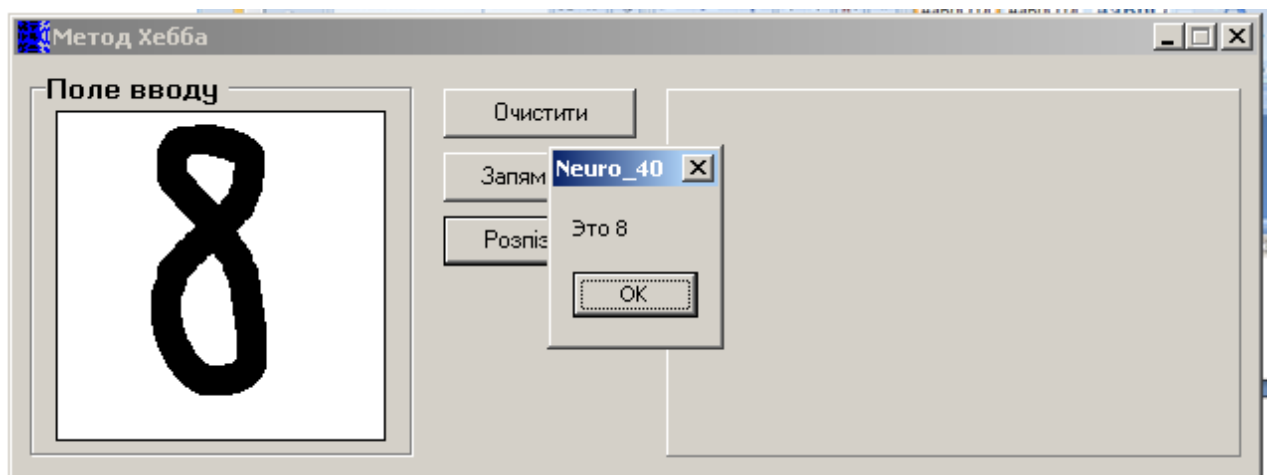


Рисунок 3.12 - Розпізнавання цифри «8»

Висновок програми вірний.

Таким чином, використовуючи простий метод Хебба ми отримали вірну та не вірну відповідь.

Застосуємо модифікований метод Хебба змінних ваг.

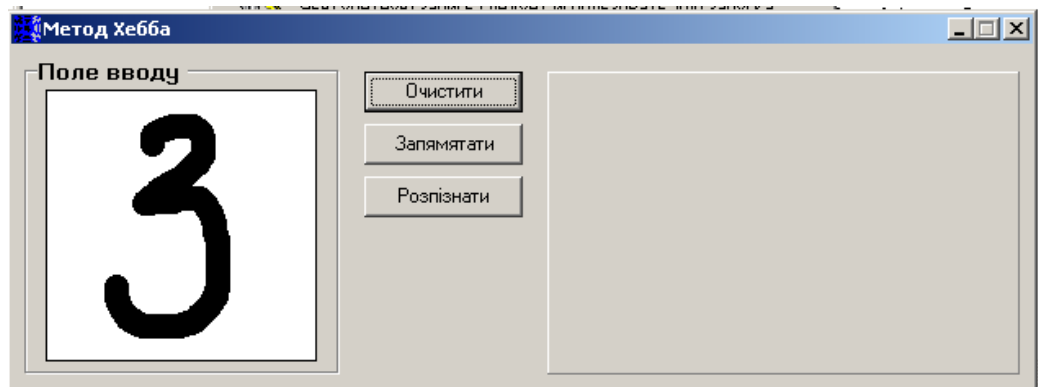


Рисунок 3.13 - Ввід цифри «3»

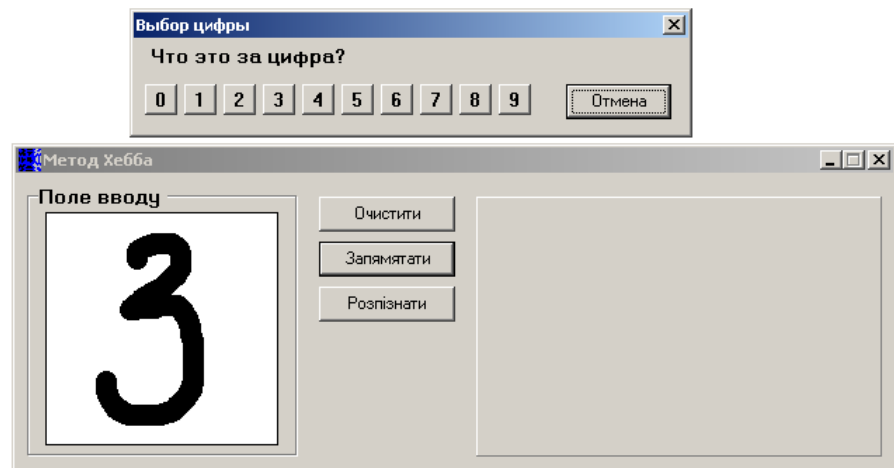


Рисунок 3.14 - Навчання цифри «3»

Аналогічні дії проводимо з цифрою «8».

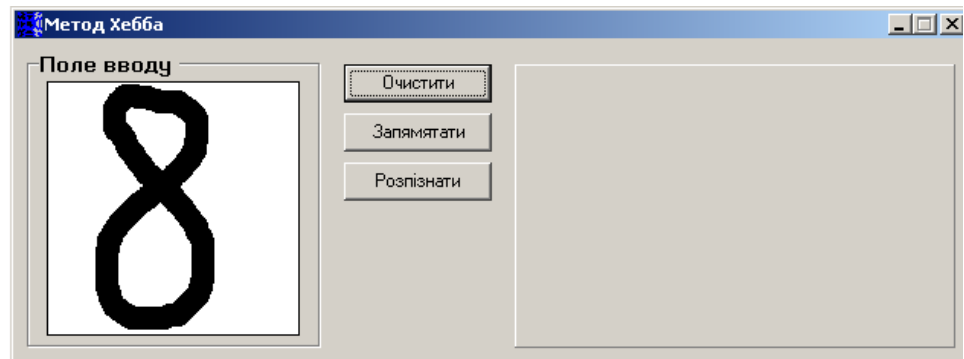


Рис. 3.12. Ввід цифри «8»

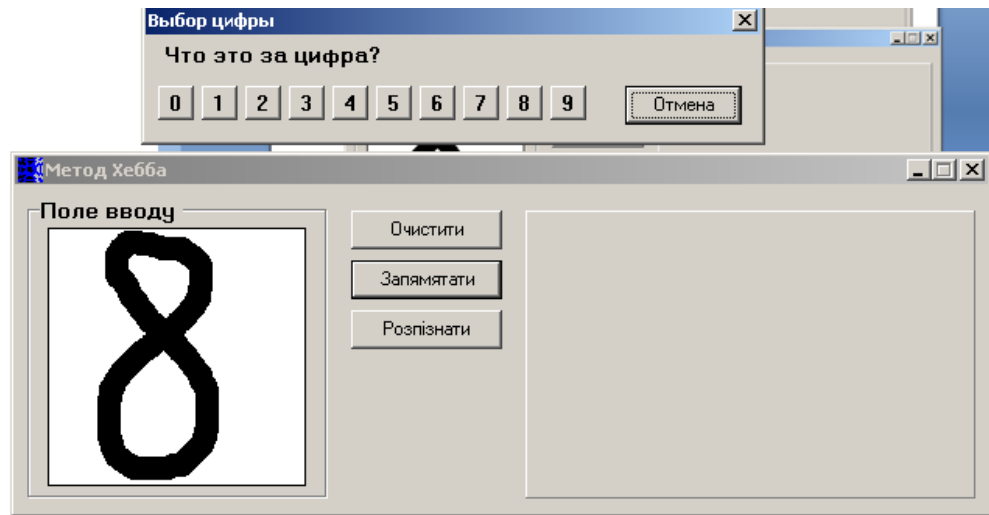


Рис. 3.13. Навчання цифри «8»

Далі, використовуючи модифікований метод Хебба розпізнаємо схожі зображення. Почнемо з цифри «3».

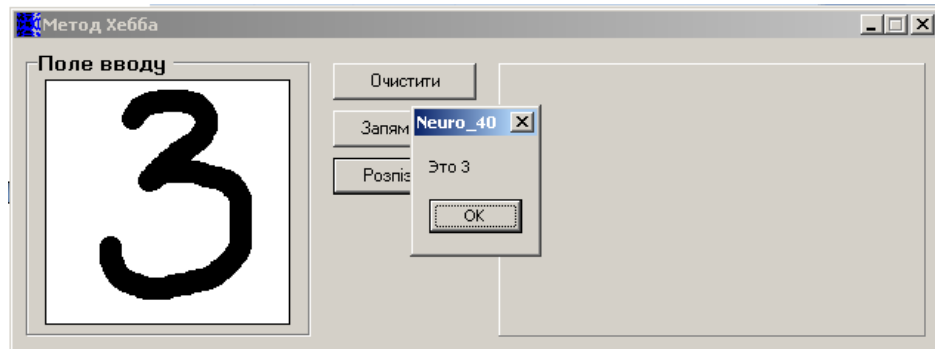


Рис. 3.14. Розпізнавання цифри «3»

Висновок програми вірний.

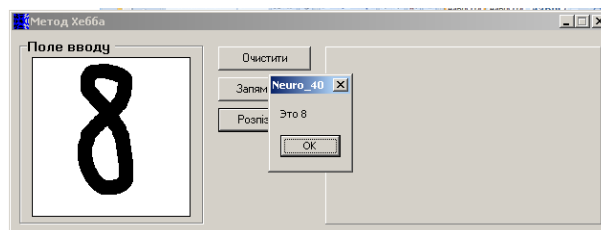


Рис. 3.15. Розпізнавання цифри «8»

Висновок програми вірний.

Таким чином, використовуючи модифікований метод Хебба ми отримали обидві вірні відповіді.

Проведемо порівняльний експеримент з 10 дослідів на розпізнавання схожих зображень простим та модифікованим методом Хебба.

№	Цифри для розпізнаван ня	Результат розпізнавання	
		Простий метод Хебба	Модифікований метод Хебба
1	«3» і «8»	Вірний/Невірний	Вірний/Вірний
2	«3» і «8»	Вірний/Вірний	Вірний/Вірний
3	«3» і «8»	Вірний/Вірний	Вірний/Вірний
4	«3» і «8»	Вірний/Вірний	Вірний/Вірний
5	«3» і «8»	Вірний/Вірний	Вірний/Вірний
6	«2» і «7»	Вірний/Вірний	Вірний/Вірний
7	«0» і «6»	Вірний/Невірний	Вірний/Вірний
8	«0» і «9»	Вірний/Вірний	Вірний/Вірний
9	«5» і «6»	Вірний/Вірний	Вірний/Вірний
10	«8» і «9»	Невірний/Невірний	Вірний/Невірний

Таблиця 3.1 - Результати експериментів

Таким чином, використовуючи простий метод Хебба ми отримали 16 вірних відповідей з 20-ти (80%).

Використовуючи модифікований метод Хебба ми отримали 19 вірних відповідей з 20-ти (95%).

Дослідження показали, що для розпізнавання схожих зображень кращу ефективність показує модифікований метод Хебба.

ВИСНОВКИ

У магістерській дисертації представлено результати, що є, відповідно до поставленої мети, вирішенням актуальних задач інтелектуального аналізу даних за допомогою нейро-фаззі систем та моделей, що самонавчаються. Дослідження, що були проведені, дозволили зробити такі висновки.

1. Удосконалено метод самонавчання тришарової автоасоціативної нейронної мережі для зниження розмірності простору ознак даних шляхом введення поліноміальної функції активації, а також запропоновано модель цієї нейронної мережі. Нейронна мережа з запропонованим методом самонавчання має можливість працювати за умов нелінійності і нестационарності вхідних даних, проте складається виключно з лінійних елементів, за рахунок чого, розроблений метод самонавчання має підвищену швидкодію.

2. Удосконалено методи самонавчання нейронних мереж для обчислення головних компонент, що відповідають максимальному і мінімальному власному значенню та їх довільної кількості, що мають підвищену швидкодію та забезпечують, на відміну від стандартних процедур, компроміс між слідкуючими та згладжуючими властивостями.

3. Удосконалено нейромережеві BSB- та GBSB-моделі, за рахунок введення нечітких функції належності, що дозволило розширити функціональні можливості при вирішенні задач автоасоціації та кластеризації в умовах класів, що перетинаються, а також синтезовано адаптивні методи самонавчання BSB- і GBSB-нейромоделей, що дозволяють підвищити швидкість налаштування мережі.

4. Вперше запропоновано модель нейромережевої автоасоціативної пам'яті на основі нечітких базисних функцій, яка характеризується підвищеною ємністю та простотою реалізації за рахунок використання одновимірних функцій належності, що дозволило зв'язати процес відновлення в нейромережевій моделі з процедурами нечіткої кластеризації.

Список використаних джерел

1. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Классификация и снижение размерности. - М.: Финансы и статистика 1989. - 607 с.
2. Алиев Р.А., Абдикеев Н.М., Шахназаров М.М. Производственные системы с искусственным интеллектом.- М: Радио и связь, 1990. - 264 с.
3. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы: Учебник.- М.: Финансы и статистика, 2004.- 424 с.
4. Базы данных. Интеллектуальная обработка информации / В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх.-М.: Нолидж, 2000, 352 с.
5. Бондарев В.Н., Аде Ф.Г. Искусственный интеллект.- Севастополь: СевНТУ, 2002.- 615 с.
6. Борисов А.Н., Алексеев А.В., Меркурьева Г.В. и др. Обработка нечеткой информации в системах принятия решений.- М: Радио и связь. 1989. - 304 с.
7. Борисов А.Н., Крумберг О.А., Федоров И.П. Принятие решений на основе нечетких моделей. Примеры использования.- Рига:Зинатне, 1990.- 184 с.
8. Васильев В.И. Распознающие системы: справочник. - К.: Наукова думка, 1983. - 423 с.
9. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности Х.: Основа, 1997.- 112 с.
10. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем.- С.-Пб.: Питер, 2001.-384 с.
11. Герасимов Б.М., Дивизинюк М.М., Субач И.Ю. Системы поддержки принятия решений: проектирование, применение, оценка

ефективності.- Севастополь: НИЦ ВСУ "Государственный океанариум", 2004.- 320 с.

12. Джонс М.Т. Программирование искусственного интеллекта в приложениях / Пер. с англ. Осипов А.И.-М.: ДМК Пресс, 2004.- 312 с.

13. Дубровін В.І., Субботін С.О. Методи оптимізації та їх застосування в задачах навчання нейронних мереж: Навчальний посібник.-Запоріжжя: ЗНТУ, 2003.-136 с.

14. Дубровин В.И., Субботин С.А., Богуслаев А.В., Яценко В.К. Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей: Монография.-Запорожье: ОАО "Мотор-Сич", 2003.- 279 с.

15. Дюк В., Самойленко А. Data mining: учебный курс.-СПб.: Питер, 2001.- 368 с.

16. Зайченко Ю.П. Основи проектування інтелектуальних систем. Навчальний посібник.- К.: Слово, 2004.- 352 с.

17. Искусственный интеллект - основа новой информационной технологии / Поспелов Г.С. - М.: Наука, 1988. - 288 с.

18. Искусственный интеллект. В 3х кн. Кн.2. Модели и методы. Справочник / Под ред. Д.А. Поспелова. - М.: Радио и связь, 1990.-304 с.

19. Каллан Р. Основные концепции нейронных сетей.- М.: Издательский дом "Вильямс", 2001.- 287 с.

20. Кричевский М.Л. Интеллектуальные методы в менеджменте.- СПб.: Питер, 2005.- 304 с.

21.Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. - М.: Горячая линия - Телеком, 2001. - 382с.

22.Кузин Л.Т. Основы кибернетики: В 2-х т. Т. 2. Основы кибернетических моделей. М.: Энергия, 1979. - 584 с.

23.Леоненков А.В. Нечеткое моделирование в среде MATLAB. - СПб.: БХВ-Петербург, 2003. - 736 с.

24. Лорьер Ж.-Л. Системы искусственного интеллекта: Пер. с франц.-М.: Мир, 1991.- 568 с.